ILLUSTRATION OF FUNDAMENTALS OF VIBRATIONS USING COMPUTABLE DOCUMENT FORMAT

by

SHIVA DESHMUKH

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

MAY 2012

ACKNOWLEDGEMENTS

I would first like to thank my thesis advisor Dr. Kent Lawrence for his expert advice and help throughout my thesis. He has always been there for me. My sincere appreciation goes to Dr. Seiichi Nomura for helping me in understanding the basics of computations and answering all my questions. I would also like to thank Dr. Bo Wang for taking the time to serve on my committee.

Finally I thank my parents, sister for their constant love and support. To all my friends for the encouragement they gave without which this was not possible.

April 17, 2012

ABSTRACT

ILLUSTRATION OF FUNDAMENTALS OF VIBRATIONS USING COMPUTABLE DOCUMENT FORMAT

Shiva Deshmukh, MS

The University of Texas at Arlington, 2012

Supervising Professor: Kent Lawrence

The main objective of this study is focused on the better understanding of the basic concepts of mechanical vibrations and its free and ready availability to the entire student community all over the globe.

In this study the concepts of vibrations are projected in the form of mathematical models. The mathematical models are designed using MATHEMATICA. These models are graphical in nature and can be animated at will. They are compiled and converted to the Computable Document Format referred as CDF.

A very special function Manipulate is used in making the models. This function has given the models an ability to compute and present various combinations which would occur on changing set of inputs. The need of reentering the inputs and recalculating them is eliminated by using Manipulate. The range in change of inputs is pre-defined.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

NOMENCLATURE

m = Mass of the object.

k = Stiffness of the spring.

c = Damping.

$u_{,} y, z, V$ = Displacement.

$\dot{u}$ = First order derivative of u i.e. velocity.

$\ddot{u}$ = Second Order derivative of u i.e. acceleration.

t = Time.

$p(t)$ = Force with respect to time.

f_s = Force corresponding to stiffness of spring.

f_d = Force corresponding to damping.

W = Weight of the block.

θ = Angle from the vertical.

I = Moment of inertia.

τ = Torque.

$L =$ Length.

$g =$ Gravitational Constant.

$C_1 = C_2 = C_3 = C_4 =$ Constants.

$\rho =$ Density.

$E =$ Young's modulus of elasticity.

$\omega =$ Frequency.

CHAPTER 1

INTRODUCTION

Vibrations have been a subject of interest since centuries. The application of this field is very vast. It ranges from Biology to Physics to Mechanics etc. It governs the Universe. Vibrations can be desirable at times as in music instruments, loudspeakers as well as tuning forks. Mostly they are unaccepted since they increase energy inputs and make undesirable sound "noise". Examples of this phenomenon can be engines, electric motors and many other mechanical devices. Weather one has to increase or create vibration or may be to end or reduce it, in any scenario deep knowledge of the field is required. To accomplish this goal understanding of the basic fundamentals of vibration in details is compulsory. It has been always challenging to come up with such a tutorial.

Various innovations have been tried to make a tutorial more understanding in nature. Providing images graphs, tables fall under this category. They really boost one's capacity in understanding a concept. With advent of new technology these boundaries can be pushed further. With the use of proper softwares it is possible to make a interactive illustration of the concept making it possible of the user to take part in the learning process. This will definitely increase the understanding as well as incorporate curiosity to try and learn new things. Such tutorials will not only be able to give the required information but also will be capable of giving extra knowledge without wasting the space and time. One of the problems faced by such a tutorial is distribution. Mass distribution of these tutorials is not possible due the requirements of the softwares they are made in. Reasons may be costs, copyrights, system configuration or any other policies.

1.1 Literature Review

There has always been lot of research and changes in the area of teaching. It can be said that many have strived hard to make masses understand concepts in simpler language. The ambition of every attempt was to convey proper idea in simplest possible way. Making bases for advanced knowledge. The preliminary ways can be said 'the traditional ways'.

Historically, the primary educational technique of traditional education was simple oral recitation. Thus books with lot of text and equations were introduced. These books had more stress in learning than understanding. This system forced to learn than to understand. It incorrectly assumes that for every ounce of teaching there is an ounce of learning by those who are taught. However, most of what we learn before, during, and after attending schools is learned without it being taught to us. A child learns such fundamental things as how to walk, talk, eat, and dress, and so on without being taught these things. Adults learn most of what they use at work or at leisure while at work or leisure. Most of what is taught in classroom settings is forgotten, and much of what is remembered is irrelevant [9]. All these things have one thing in common that is learning depends on what we observe and what we experiment. A different approach was taken in order to eliminate the traditional method and also to enhance quality of education.

The new techniques had various advantages over the traditional method. It uses modern gadgets and computers to give out ideas. E-books, power point presentations, videos etc are the few examples of these new techniques. E-books eliminate the possibility of carrying huge books around. They also have active links to references which makes it possible for users to browse if necessary. Power point pretentions popularly known as "PPTs" have different approach. They work on better association and presentation to make understand a point. They are found to be more effective than E-books. They are widely used as a effective teaching tool. The most effective way to teach has always been though eyes. It is said that understanding of any concept is better if it is visualized. Thus showing a video about a topic will help in maximum understanding. The only flaw in all these techniques is they are premade. An aspirant can only see what is shown the point being they cannot change the conditions or focus on a particular area. The imagination power of the user is ceased. To eliminate these drawbacks one more step is taken. Applets using java are being made.

One of the current works on java applets for Mechanical Engineering is undergoing at Purdue University. The goal of this project is to develop interactive, animated Java applets to demonstrate fundamental concepts for use in fluid mechanics, mechanics, and vibration courses in mechanical engineering. The applets are expected to increase student comprehension of fundamental concepts presented in the courses where the applets are used. The inconvenience in creating these applets is the use of java language. Using java to make such kind applets can be very complex. It may require thousands of lines of code to design a simple applet. Size of the file can also be an issue at times. Moreover the system configuration on which the applet is played plays an important role for proper functioning of the applet. A tool which fits above criterion and eliminate the defects should be used.

1.2 Mathematica & CDF

Mathematica is such a tool which can be used instead of java. Mathematica is a computational software program used in scientific, engineering, and mathematical fields and other areas of technical computing created by Wolfram Research . It is known as world's best application for computation. Mathematica has vast number of advantages over other tools.

Mathematica applies intelligent automation in every part of the system, from algorithm selection to plot layout and user interface design to get reliable, high-quality results without needing algorithm expertise. Another important advantage is the programming language. Mathematica stands out from traditional computer languages by simultaneously supporting many programming paradigms, such as procedural, functional, rule-based, pattern-based and more. Mathematica is unique among technical computing platforms because it includes an enormous collection of carefully assembled data of all kinds, continuously updated and expanded. It also has various built in functions which can be recalled as required. These mathematical functions save both time and labor as there is no need to write codes each time they are needed. A free-of-charge version, Wolfram CDF Player, is provided for running Mathematica programs that have been saved in the Computable Document Format (CDF).

CDF is an electronic document format designed to allow easy authoring of dynamically generated interactive content. It was created by Wolfram Research. Computable document format supports GUI (graphical user interface) elements such as sliders, menus and buttons. Content is updated using embedded computation in response to GUI interaction. Contents can include formatted text, tables, images, sounds and animations. CDF supports Mathematica typesetting and technical notation. Paginated layouts, structured drill down layout and slide-show mode are supported. Styles can be controlled using a cascading style sheet. It create content as everyday as a document, but as interactive as an application. CDF files can be read using a free CDF Player which can be downloaded from Wolfram Research. CDF reader support is available for Microsoft Windows, Macintosh and Linux. All these above points make CDF a great tool for distribution of files [8].

1.3 Research Objectives & Outline

The main objectives of this study are:-

- To create an interactive document with animation to display fundamentals of Mechanical Vibration.
- To convert it into a certain format, this can be distributed freely and can be used without buying any software.

The thesis will have the following sections:-

Chapter 2 will discuss single degree of freedom problems and parameters which affect them. The nature of the effect of various inputs on single degree of freedom is shown. It will also cover mode shapes for beams under different conditions.

Chapter 3 will throw light on multiple degree of freedom problems. Multiple inputs are considered. Beat phenomenon is studied in detail.

Chapter 4   contains model like quarter car model and the vibration absorber.

Chapter 5   has results and conclusions.

CHAPTER 2

SINGLE DEGREE OF FREEDOM

A single degree of freedom system is the simplest mechanical system possible. It can move by translation along one direction only, or can rotate about one axis. The motion of a single degree of freedom system is a sinusoidal having only a single frequency. Mechanical structures are always more complex than the single degree of freedom system, but they can be thought of a being built up of a collection of single degree of freedom systems. This is somewhat analogous to a complex wave form being considered as a collection of sinusoidal components. The disciplines of modal analysis and finite element modeling treat mechanical systems in this way, and the number of degrees of freedom they possess determines their complexity [10].

Assumptions made during calculations:-

- Only horizontal motion is considered.
- Internal damping and mass of the spring is neglected.
- Air resistance is neglected.
- Displacements are measured from static equilibrium position.

2.1 Spring Mass System

The most important and classical example is one with simple spring and a block popularly known as mass-spring-dashpot system. It can be considered as symbolic representation of almost any single degree of freedom problem in real life.

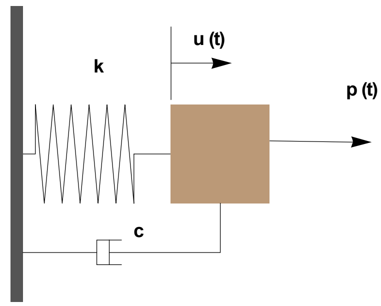


Figure 2.1 Mass-spring-dashpot system.

As show in figure 2.1 the system consists of a block which is attached to a spring with a damping system. Whole system is attached to rigid base. In order to find equation of motion for this system it is important to take a look at free-body diagram.

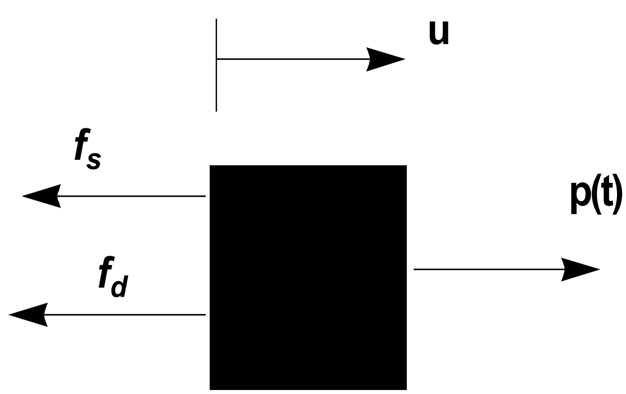


Figure 2.2 Free body diagram.

Appling Newton's Second Law of Motion to the above system.

$$\sum F_x = m\ddot{u} \tag{2.1.1}$$

(Forces in +ve x direction are considered to be positive)

$$p(\mathrm{t}) - f_s - f_d + W = m\ddot{u} \tag{2.1.2}$$

using the relation between motion variables and forces:-

$$f_s = ku \tag{2.1.3}$$

$$f_d = c\dot{u} \tag{2.1.4}$$

Combining and rearranging the terms from the equations

$$m\ddot{u} + c\dot{u} + ku = W + p(t) \tag{2.1.5}$$

Adjusting displacement measured relative to the static equilibrium position above equation can be written as

$$m\ddot{u} + c\dot{u} + ku = p(t) \tag{2.1.6}$$

Equation 2.1.6 is the fundamental equation in liner vibration theory. Almost all examples of SDOF can be described by different forms of this equation [1]. By calculating the roots of this equation the motion of the block can be predicted. With variance in the values of m, c, k or the force the motions of the block differs. Calculation of these roots is done using Mathematica. Variance can be caused by using function 'Manipulate'. This helps us visualize the changes in the motion with respect to the variables.

Following graphs represents the displacement of the block with respect to a set of values for these variables. It has no damping small initial displacement and no external force acting on it.

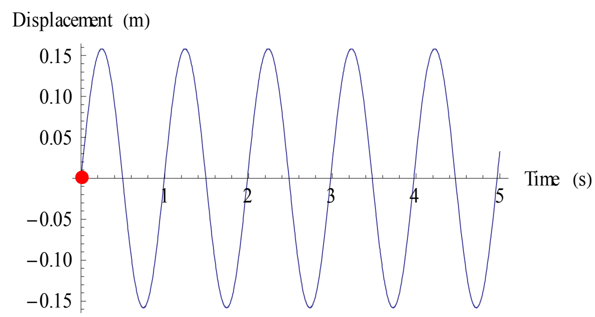


Figure 2.3 Graph u(t) vs. t no damping.

Motion of the block remains a constant cycle. The dot present follows the displacement with time. If the initial boundary conditions are changed so changes the nature of the graph. For example if the initial displacement is increased and damping is included also if the block is given initial velocity the graph will look as follows.

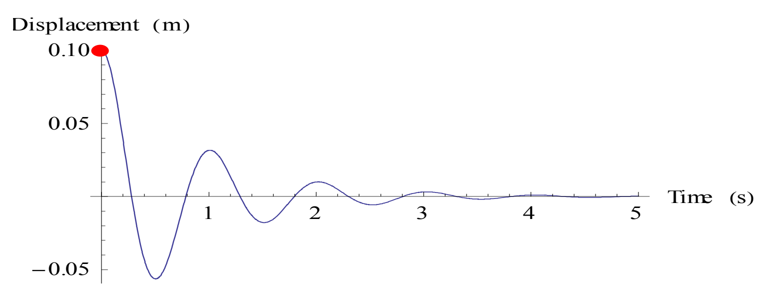


Figure 2.4 Graph u(t) vs. t with damping.

It can be seen that displacement reaches a maxima at starting of the motion. As the time progresses the displacement reduces and eventually dies out. Once the provided codes are utilized they will animate the position of the block throughout the time frame for given set of inputs. Once the animation runs out changes to input can be made using the tabs provided and the animation can again be used to demonstrate the position of the block.

2.2 Various Force Inputs for Spring Mass System

It is very interesting to observe the effect of various input forces on a single degree of freedom system like the mass-spring-dashpot system. It helps us understand a relation between the applied force and the displacement of the block so caused. The response is normalized by dividing the displacement with division of amplitude of applied load and stiffness of spring. Various inputs that considered are:-

- Step force input.
- Rectangular pulse force input.
- Ramp force input.
- Triangular pulse force input.

2.2.1 Step Force Input

As the name suggest the applied force is always kept constant throughout the time frame. The magnitude of the force can changed within a stipulated range defined by the programmer. Changes in various input conditions can also be made. The nature of the graph with constant input is show in the figure 2.5.

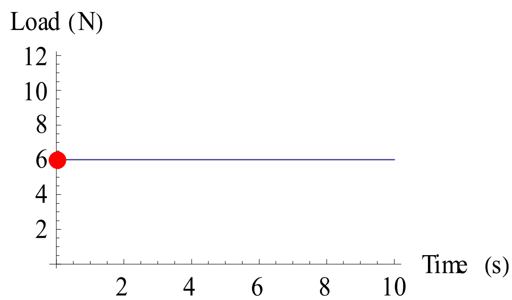


Figure 2.5 Graph force vs. time step input.

The required value of initial force is substituted in equation 2.1.6 which is then solved using a Mathematica command 'NDSolve'. The roots of the equation are used to plot the displacement plot as well as they are used to animate the block. For given problem with constant force and certain values of m, k and c the displacement graph looks almost the same as in with no input but displacement will not die. It acquires a constant value and will remain on that value till the end of time frame.

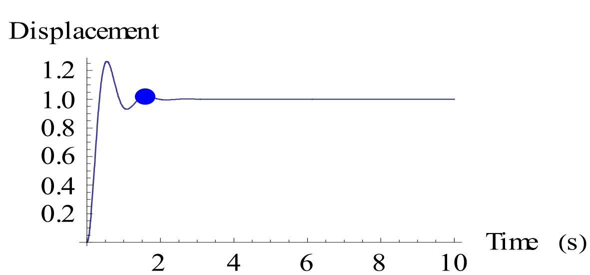


Figure 2.6 Graph u(t) vs. t step input.

2.2.2 Rectangular Pulse Force Input

An input whose magnitude is fixed for some time after which it is reduced to zero or any other constant lower or higher value will be called as a step input. Another reason to call it a step input is the nature of its graph looks like a step. To obtain such input a small condition is introduced in Mathematica.

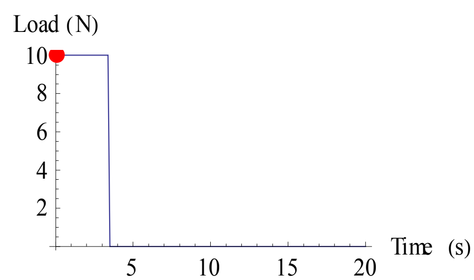


Figure 2.7 Graph force vs. time rectangular pulse input.

Motion of the block caused by such input is very interesting in nature. Displacement graph revels that the block tries to attain a constant displacement value with the initial load but as the input becomes zero the above case is similar to the system with no load and initial displacement. The displacement then falls and tries to attain a zero value. The values of damping constant and the stiffness of the spring decide dying of the vibrations in such case.

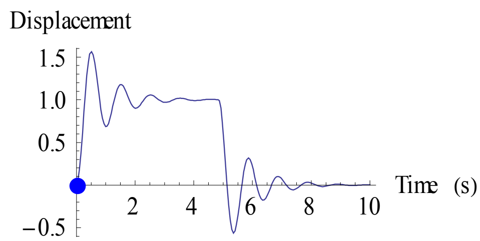


Figure 2.8 Graph u(t) vs. t rectangular pulse input.

It is clear from the above graph that after time 5 units it looks exactly the same as in graph from figure 2.4. Also if it is observed till time 5 units it looks similar to figure 2.6 making it combination of both conditions.

2.2.3 Ramp Force Input

The magnitude of the input load increases as the time increase. The nature of increment of the load is constant throughout the time frame. This can be achieved in Mathematica by associating the applied force with the time itself. The nature of the graph is like that of a slant straight line with positive slope.

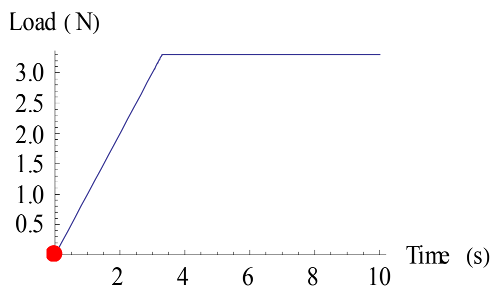


Figure 2.9 Graph force vs. time ramp input.

Application of such an input load on the mass-spring-dashpot system makes the mass move with increasing displacement. It is very difficult to stabilize such systems. Under such conditions it's only possible to reduce the magnitude of displacement. It might be possible to make the vibrations die by providing other inputs which vary according to the applied force.

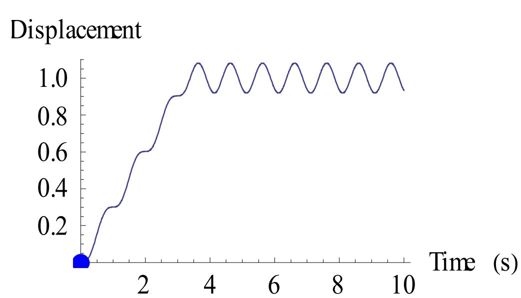


Figure 2.10 Graph u(t) vs. t ramp input.

2.2.4 Triangular Pulse Force Input

In this type of input magnitude of the load increases with time up to a certain value after which it decreases with time. This make the graph of input verses time look like a triangle. The input load which is used here is made to look like a equilateral triangle. This achieved by increasing the load gradually till the time reaches its middle value after which the load is gradually decreased.

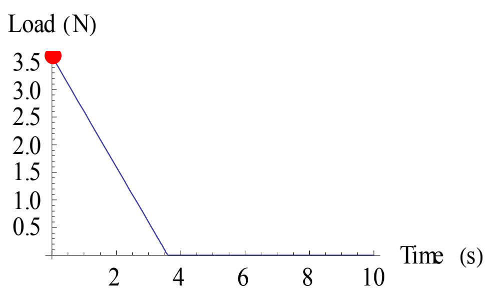


Figure 2.11 Graph force vs. time triangular pulse input.

Displacement caused by such an input is shown in the following figure.

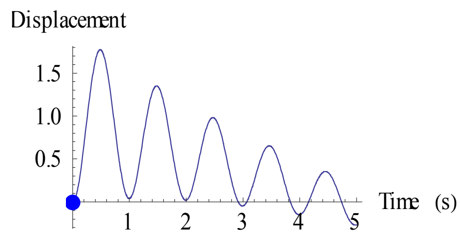


Figure 2.12 Graph u(t) vs. t triangular pulse input.

2.3 Moving Base for Spring Mass System

The rigid base to which mass-spring-dashpot system is attached is given a vibration of known frequency. Vibrations due to such an unstable base are then studied.

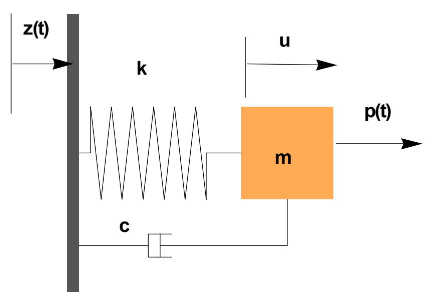


Figure 2.13 Mass-spring-dashpot system with moving base.

As shown in the figure 2.13 base is moving with z (t). It is assumed that motion of the base is known. When the spring is not stretched $u = z = 0$. In order to derive the equation of motion for the block free body diagram is drawn.

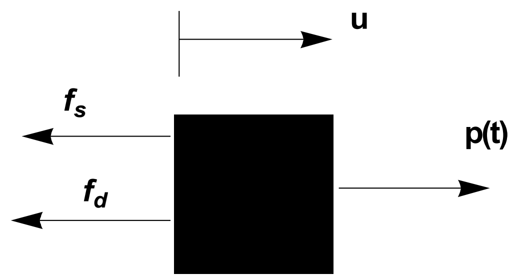


Figure 2.14 Free body diagram.

Applying Newton's Law of motion and using dynamic equilibrium equation.

$$\sum F_x = m\ddot{u} \qquad (2.3.1)$$

From free body diagram,

$$p(t) - f_s - f_d - m\ddot{u} = 0 \qquad (2.3.2)$$

Relating the motion variables to forces and simplifying

$$m\ddot{u} + c(\dot{u} - \dot{z}) + k(u - z) = p(t) \qquad (2.3.3)$$

Writing the known quantities on right-hand side

$$m\ddot{u} + c\dot{u} + ku = c\dot{z} + kz + p(t) \qquad (2.3.4)$$

It is favorable to write motion equation in terms of the displacement of the mass relative to moving base. Let,

$$w = u - z \qquad (2.3.5)$$

Substituting equation 2.3.5 to equation 2.3.4

$$m\ddot{w} + c\dot{w} + kw = p(t) - m\ddot{z} \qquad (2.3.6)$$

If equation 2.3.6 is observed carefully it can be seen that it is similar to equation 2.1.6 [1]. Conclusions can be drawn that due to base motion a reverse inertial force $m\ddot{z}$ is being added. If no external force acts on the system and the damping constant is reduced to zero the motion of the block will be as follows:-

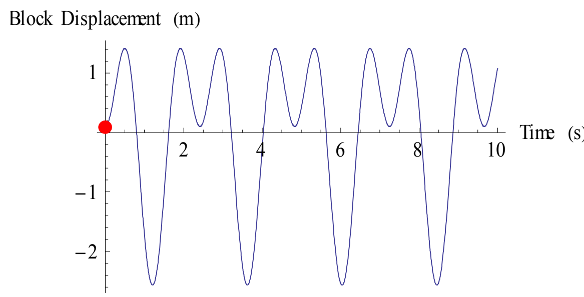


Figure 2.15 Graph u(t) vs. t mass-spring-dashpot system with harmonic base motion.

Input conditions are slightly changed. Block is given an initial displacement. Damping constant is added. Stiffness of the spring is increased. Base excitation frequency is also increased. It is observed that displacement of the block takes a constant cycle after a small passage to time.

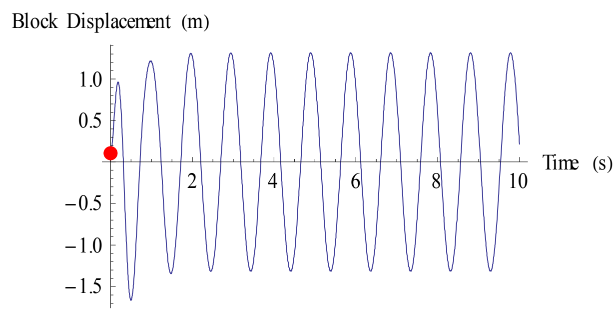


Figure 2.16 Graph u(t) vs. t mass-spring-dashpot system with harmonic base motion.

2.4 Non Linear Pendulum System

Pendulum can be defined as a mass attached to an end of a rod or a string whose other end is connected to a pivot so as the system can swing freely generally under the influence of the gravitational force. Some assumptions are made while considering a pendulum system.

- Mass is considered as concentrated at the center.
- Pivot is frictionless.
- Rod is weightless.

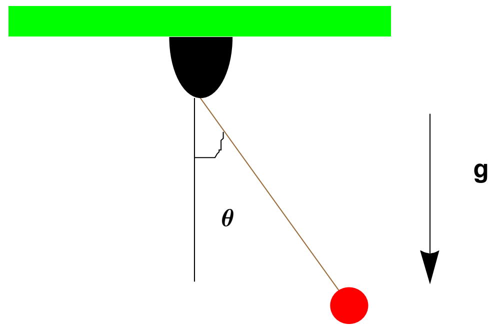


Figure 2.17 Non linear pendulum system.

The derivation of the equation of motion for the above system is very simple.

Appling Newton's second law in rotational form,

$$\tau = I\ddot{\theta} \tag{2.4.1}$$

For pendulum torque is given by

$$\tau = -mgLSin(\theta) \tag{2.4.2}$$

Moment of inertia is

$$I = mL^2 \tag{2.4.3}$$

Combining all the above equations we get

$$\ddot{\theta} = -\frac{g}{L}Sin\theta \tag{2.4.4}$$

Equation 2.4.4 is the equation of motion for the pendulum system. The various variables that can be changed are length of the rod, initial displacement and initial velocity. The range of all these variables is pre defined in Mathematica codes. Results of combinations of these variables can be used to get the required displacement of the mass. Result of one such combination is shown below.

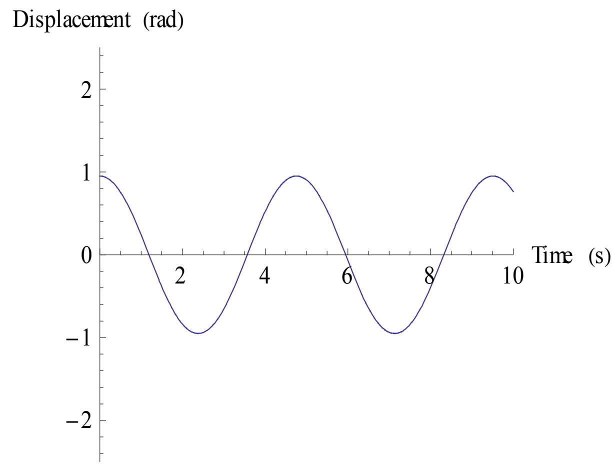


Figure 2.18 Graph displacement vs. time pendulum system.

2.5 Free Transverse Vibration of Bernoulli-Euler Beams

Natural elastic modes of vibrations for uniform beams with various end condition are studied.

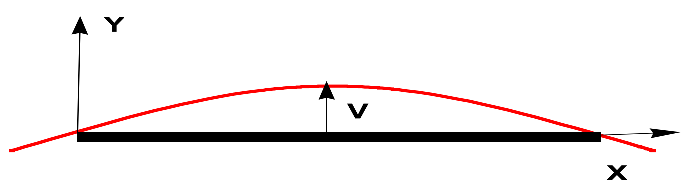


Figure 2.19 Beam notation.

The end conditions can be as following:-

Free end $$\frac{d^2V}{dx^2} = \frac{d^3V}{dx^3} = 0 \tag{2.5.1}$$

Fixed end $$V = \frac{dV}{dx} = 0 \tag{2.5.2}$$

Simply supported end $$V = \frac{d^2V}{dx^2} = 0 \tag{2.5.3}$$

Due to wide application of beams study of nature of its vibration is important. Beams are not only present in structures like buildings and bridges but also are used in turbines, generators, pumps etc. All these applications are subjected to constant vibrations thus making it a priority to find its effects on beams.

Figure 2.20 Simply supported beam.

Let us first consider a uniform beam with simply supported conditions. It's well known how the general solution for this case is derived. Making use of the above said general solution we have

$$V(x) = C_1 \sinh \lambda x + C_2 \cosh \lambda x + C_3 \sin \lambda x + C_4 \cos \lambda x \tag{2.5.4}$$

According to boundary conditions, at x=0;

$$V(0) = 0, \Rightarrow \quad \frac{d^2V}{dx^2} = 0 \tag{2.5.5}$$

At x=L;

$$V(L) = 0, \Rightarrow \quad \frac{d^2V}{dx^2} = 0 \tag{2.5.6}$$

Taking double derivative of equation 2.5.4

$$\frac{d^2V}{dx^2} = \lambda^2 (C_1 \sinh \lambda x + C_2 \cosh \lambda x - C_3 \sin \lambda x - C_4 \cos \lambda x) \tag{2.5.7}$$

Evaluating the boundary conditions at x=0 we get $C_2 = C_4 = 0$, applying other conditions;

$$C_1 \sinh \lambda L + C_3 \sin \lambda L = 0 \tag{2.5.8}$$

$$\lambda^2 (C_1 \sinh \lambda L - C_3 \sin \lambda L) = 0 \tag{2.5.9}$$

In order that the above liner equations 2.5.8 and 2.5.9 should have a nontrivial solution their determinant must be equal to zero. Simplifying,

$$\sinh \lambda L \sin \lambda L = 0 \tag{2.5.10}$$

The only possible nontrivial solution being;

$$\sin \lambda L = 0 \tag{2.5.11}$$

Equation 2.5.11 is used to determine eigenvalues also we know that

$$\lambda^4 = \omega^2 \frac{\rho A}{EI} \tag{2.5.12}$$

Therefore equations 2.5.11 and 2.5.12 can be used to find natural frequency ω_r

$$\omega_r = \left(\frac{r\pi}{L}\right)^2 \left(\frac{EI}{\rho A}\right)^{1/2} \tag{2.5.13}$$

It can be found out that $C_1 = C_2 = C_4 = 0$ by substituting equation 2.2.11 to 2.5.8.

So the mode shapes of the beam are given by;

$$V_r(x) = C \sin \lambda_r x \tag{2.5.14}$$

If such a simply supported beam is allowed to vibrate the first four mode shapes will look as;

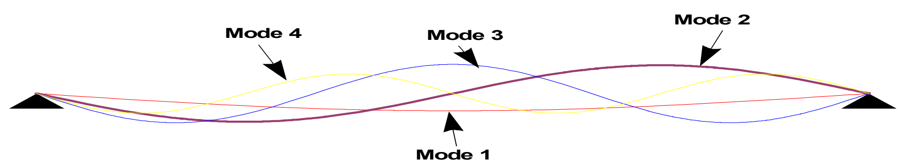
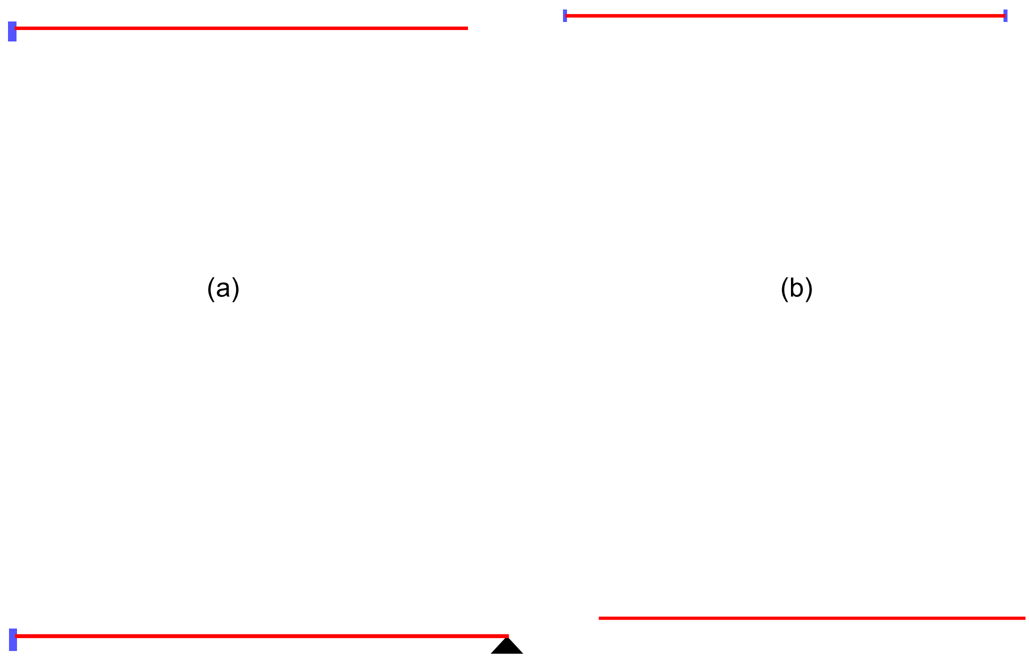


Figure 2.21 Mode shapes of simply supported beam.

In a similar ways the equations for beams with different end conditions are derived. Equations and mode shapes changes as the end conditions are changed. Four other conditions are considered and are modeled using Mathematica. Models are capable of showing different modes of a beam with a given end condition.



(a) (b)

(c) (d)

Figure 2.22 a) Cantilever beam. b) Clamped and clamped beam.
c) Clamped and simply supported beam. d) Free beam.

The mode shapes of these cases as obtained using the codes in Mathematica are shown in the following figure.

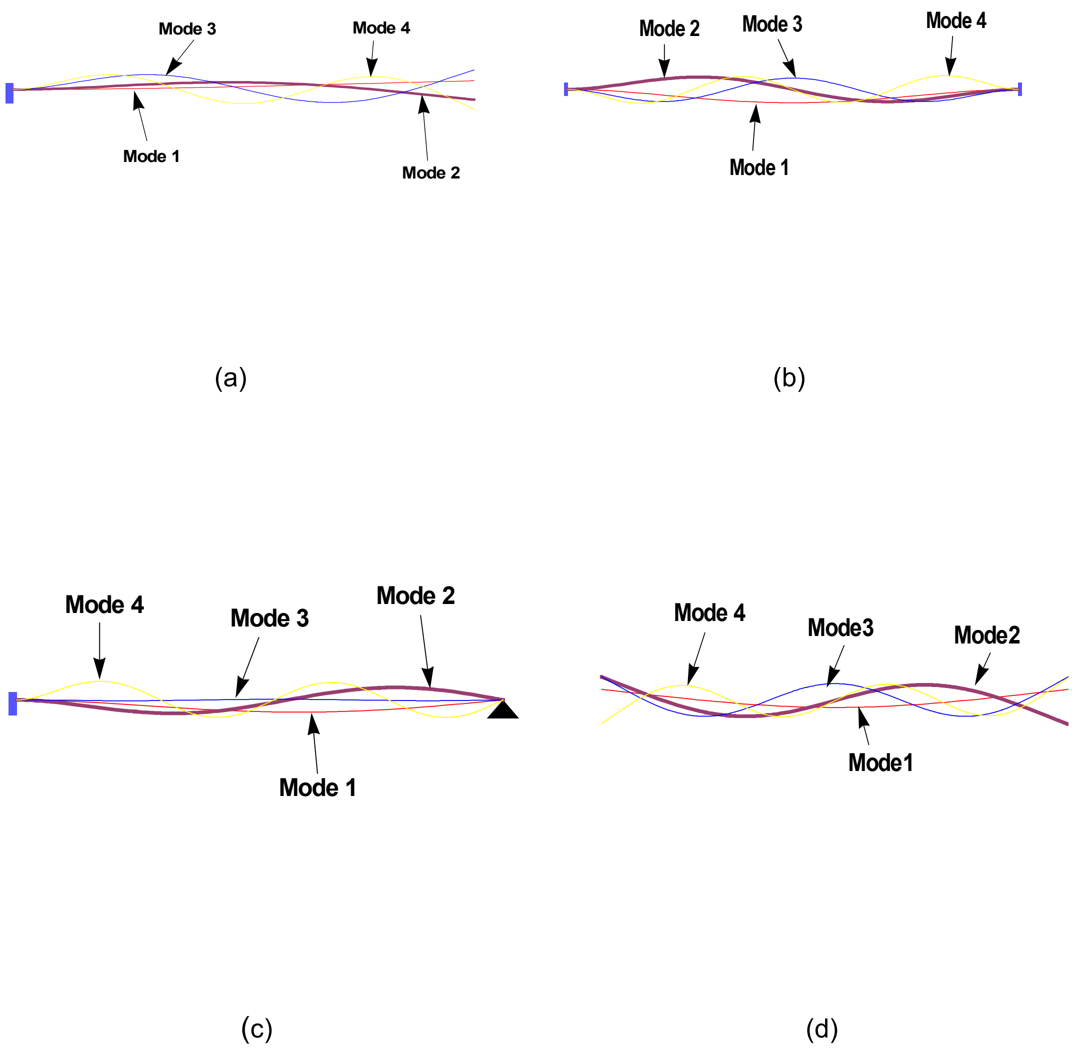


Figure 2.23 Mode shapes. a) Cantilever beam. b) Clamped and clamped beam. c) Clamped and simply supported beam. d) Free beam.

CHAPTER 3

MULTIPLE DEGREES OF FREEDOM

As stated in Chapter 2, most engineering systems are continuous and have an infinite number of degree of freedom. The vibration analysis of continuous systems requires the solution of partial differential equations which is tedious. The truth being analytical solution does not exist for many such equations. The analysis of multiple degrees of freedom for such engineering system requires the solutions of ordinary differential equations which are relatively simple. Multiple degrees of freedom can be expressed as many single degree of freedom working simultaneously.

3.1 Spring Mass Two Degrees of Freedom System

In this system a rigid base is attached with a spring and a block. A dashpot is also attached with them. Another block with spring is attached to the first block with addition of one more dashpot. The assembly is shown in the following figure;

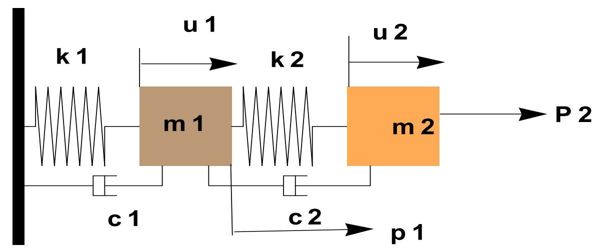
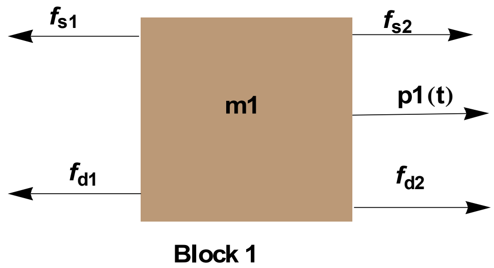


Figure 3.1 Spring mass two degrees of freedom.

Above system can be considered as combination of two separate single degree of freedom system. Equation of motion of such systems can be calculated by combining the equations of motion of the comprising single degree of freedom systems. Making use of free-body diagram of the system;



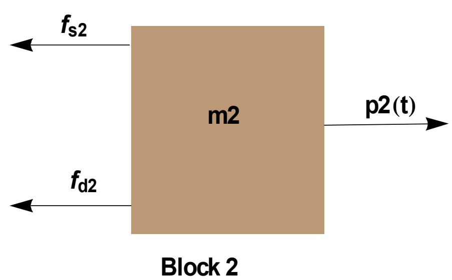


Figure 3.2 Free body diagrams.

$$\xrightarrow{+} \sum F_{x1} = m_1\ddot{u}_1 = -f_{s1} - f_{d1} + f_{s2} + f_{d2} + p1(t) \qquad (3.1.1)$$

$$\xrightarrow{+} \sum F_{x2} = m_2\ddot{u}_2 = -f_{s2} - f_{d2} + p2(t) \qquad (3.1.2)$$

Relating the liner elastic spring forces to displacement and the viscous damping forces to the velocities and simplifying equations 3.1.1 and 3.1.2.

$$m_1\ddot{u}_1 + \dot{u}_1(c_1 + c_2) - c_2\dot{u}_2 + k_1u_1 + k_2(u_1 - u_2) = p1(t) \qquad (3.1.3)$$

$$m_2\ddot{u}_2 + k_2(u_2 - u_1) + c_2(\dot{u}_2 - \dot{u}_1) = p2(t) \qquad (3.1.4)$$

Equation 3.1.3 and 3.14 are called equation of motion for block one and block two respectively.

The loads or the applied forces p1 (t) and p2 (t) are constant and are considered as zero. Both the blocks can be given initial displacement. The Mathematica code allows inputting initial displacement in negative x- direction also. This will help in understanding the effect of reversal displacement application to a single system. Other parameters which can be controlled in the code include stiffness of springs, mass of the blocks and damping constants of the dashpots. Displacements of block one and two are shown when the applied u1 (0) and u2 (0) are applied in opposite directions. Blue line represents displacement of block 1 while red line represent displacement of block 2.

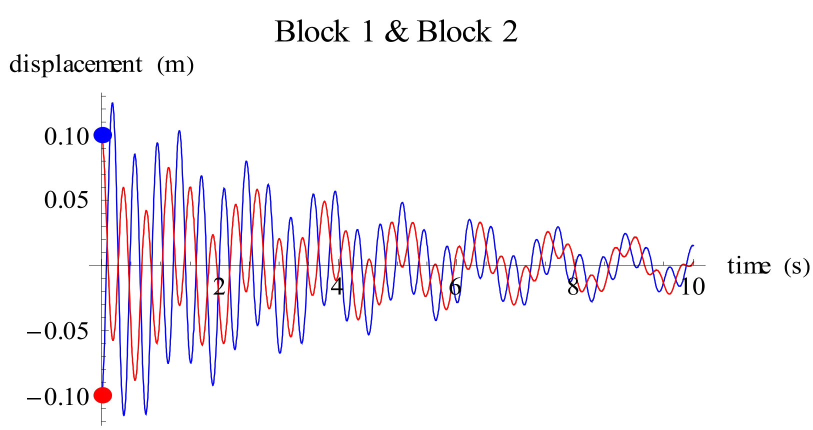


Figure 3.3 Graph u1(t),u2(t) vs. t for two degree of freedom spring-mass system.

3.2 Spring Mass Three Degrees of Freedom System

Three single degree of freedom systems are combined to make one multiple degree of freedom system. Making this assumption helps to solve very complex systems by mere repetitions of simple calculations. These breakdowns or buildups often save time and money for the complex calculation.

A shear-frame building model, shaft-disk model and a spring- mass model will have same equations of motion. Consider a spring mass model with three spring a rigid base and three blocks.

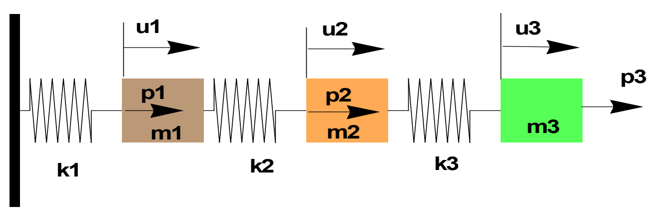


Figure 3.4 Three degrees of freedom system spring-mass model.

Aim being to find the equations of motion for the above shown system. Draw the free body diagram of each block representing all the unknown forces.

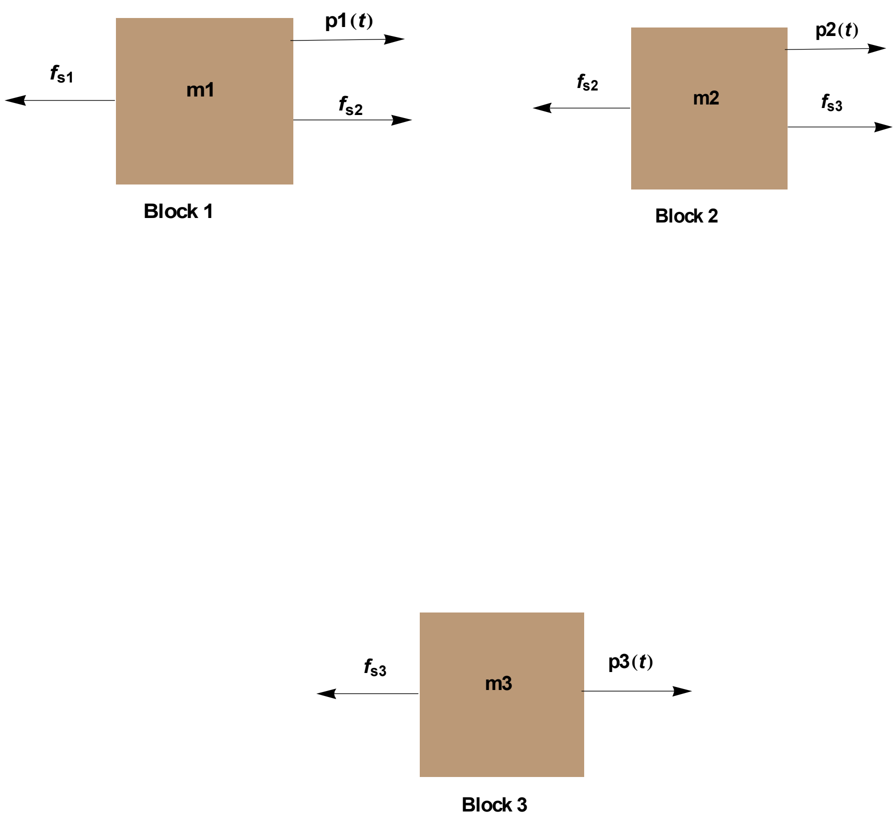


Figure 3.5 Free body diagram of three degrees of freedom system

Using Newton's second law for each block;

$$\xrightarrow{+} \sum F_{x1} = m_1\ddot{u}_1 = -f_{s1} + f_{s2} + p1(t) \qquad (3.2.1)$$

$$\xrightarrow{+} \sum F_{x2} = m_2\ddot{u}_2 = -f_{s1} + f_{s3} + p2(t) \qquad (3.2.2)$$

$$\xrightarrow{+} \sum F_{x3} = m_3\ddot{u}_3 = -f_{s3} + p3(t) \qquad (3.2.3)$$

Relating the linearly elastic spring forces to the displacements and then simplifying the above equations;

$$m_1\ddot{u}_1 + (k_1 + k_2)u_1 - k_2u_2 = p1(t) \tag{3.2.4}$$

$$m_2\ddot{u}_2 - k_2u_1 + (k_2 + k_3)u_2 - k_3u_3 = p2(t) \tag{3.2.5}$$

$$m_3\ddot{u}_3 - k_3u_2 + k_3u_3 = p3(t) \tag{3.2.6}$$

Equations 3.2.4, 3.2.5 and 3.2.6 are the equations of motion for the system shown in figure 3.2.1. They govern individual motion of each of the blocks. This system has no dampers so it will be continuously vibrating system which will not acquire any steady state. The Mathematica model of this system is similar to the previous model and has all same privileges.

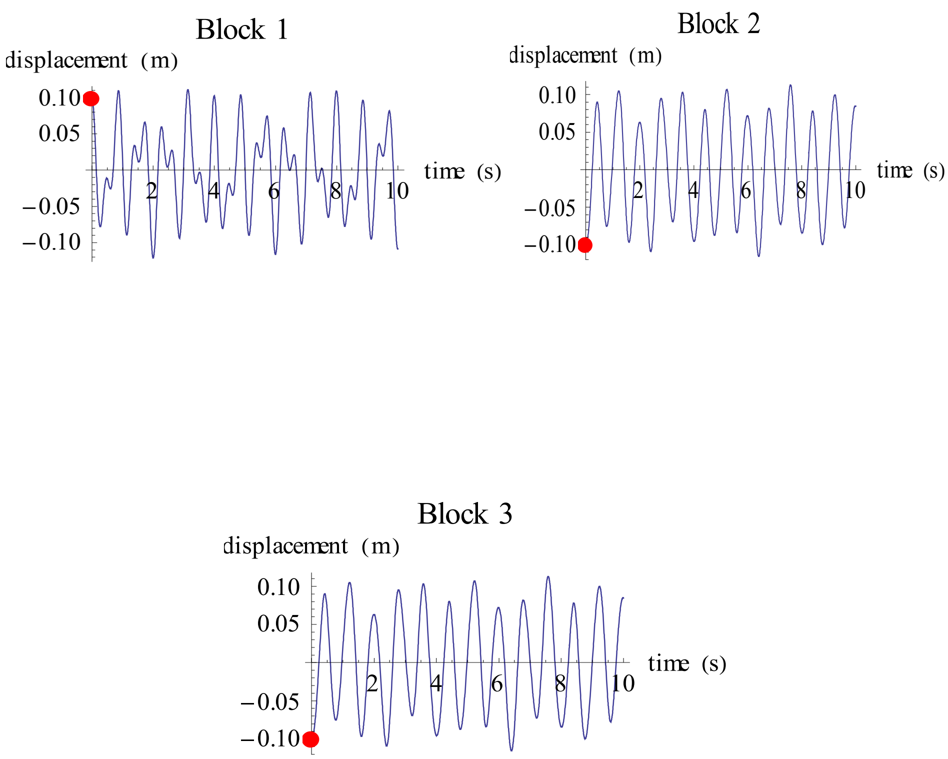


Figure 3.6 Graphs displacement vs. time three degree of freedom system.

3.3 Coupled Pendulums

This system is best to show the effect of closely placed natural frequencies which gives rise to a unique phenomenon in vibrations called as 'the beat phenomenon'. Examples of such systems include two masses which are connected by a weak spring or two identical pendulums coupled with weak spring. Transfer of energy takes place in the coupled system which could induce vibration in the primary system instead of suppressing them.

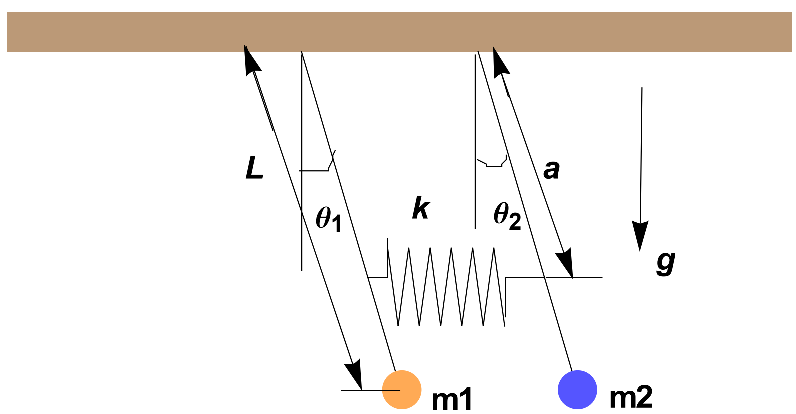


Figure 3.7 Coupled pendulums.

The Mathematica model of this system allows changing all the parameters except the length of the rods of pendulum and the length to which spring is attached. The equation of motion for the above system is derived as follows;

Free body diagram of the system.

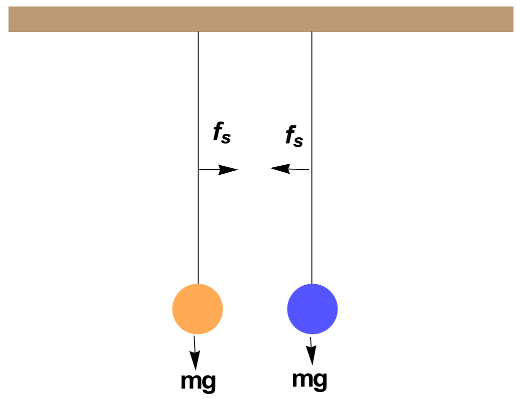


Figure 3.8 Free body diagram of coupled pendulums.

For pendulum one; assuming anticlockwise as positive,

$$\sum M = I_1 \ddot{\theta}_1 \tag{3.3.1}$$

$$-f_s a^2 + m_1 g \sin\theta_1 L = I_1 \ddot{\theta}_1 \tag{3.3.2}$$

Relating the linearly elastic spring forces to the displacements and then simplifying the above equation;

$$I_1 \ddot{\theta}_1 + (ka^2 + m_1 gL)\theta_1 - ka^2 \theta_2 = 0 \tag{3.3.3}$$

Similarly for pendulum two;

$$I_2 \ddot{\theta}_2 - ka^2 \theta_1 + (ka^2 + m_2 gL)\theta_2 = 0 \tag{3.3.4}$$

Equations 3.3.3 and 3.3.4 are equations of motion for pendulum one and two respectively.

It will be interesting to see the effect of three conditions, one when the pendulums are in phase i.e. $\theta_1 = \theta_2$. Second, the pendulums are out of phase i.e. when $\theta_1 = -\theta_2$ or vice versa. Third being the one which triggers 'the beat phenomenon'.

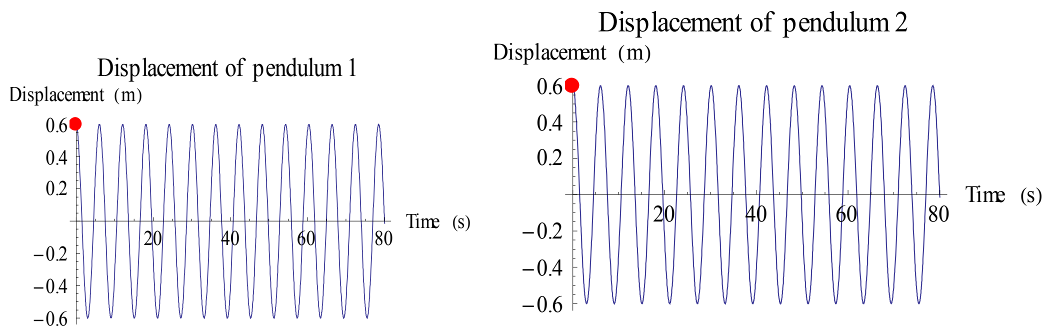


Figure 3.9 Displacement vs. time for pendulums in phase.

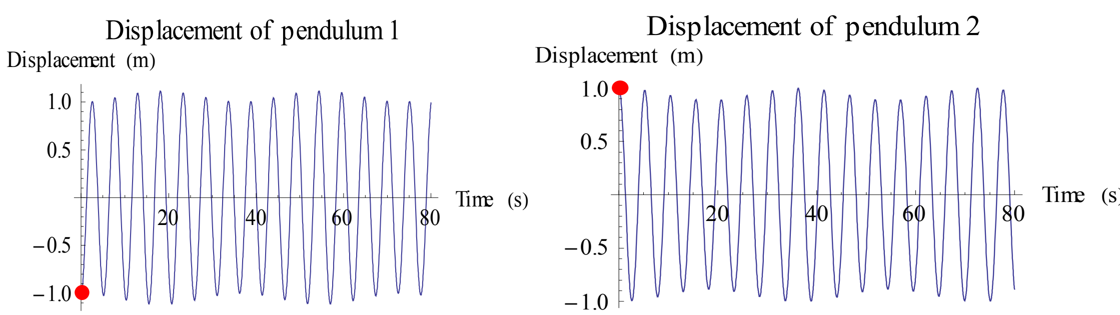


Figure 3.10 Displacement vs. time for out of phase pendulums.

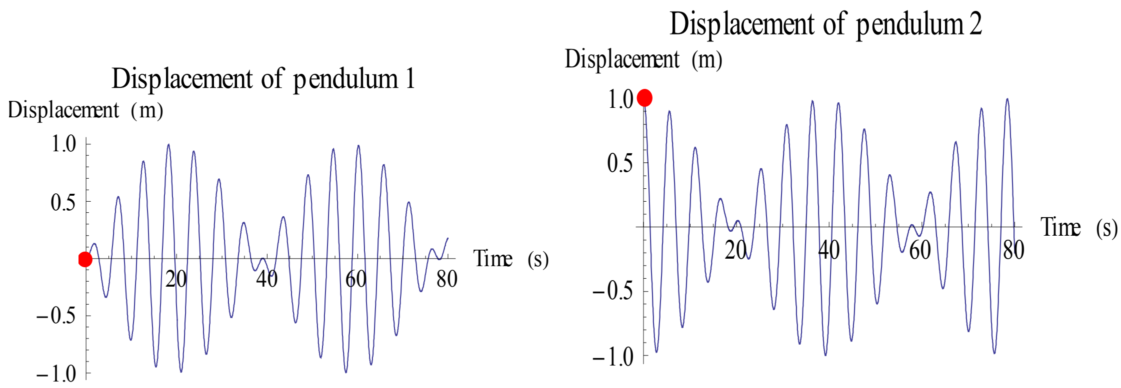


Figure 3.11 'The beat phenomenon'.

The beat phenomenon is obtained by keeping the initial displacement of pendulum one to zero while small displacement is given to pendulum two. Another condition to be fulfilled is $k << \frac{mgL}{a^2}$.

CHAPTER 4

MODELS

Simple models which can be applied in real world are covered in this chapter. All these models are created by above mentioned techniques and concepts. They are very powerful tools in their respective fields. As these models can animate so they expand imagination limits and can help in better achievements of required goals. Addition or subtraction of the parameters to vary can be done very easily.

4.1 Quarter Car Model

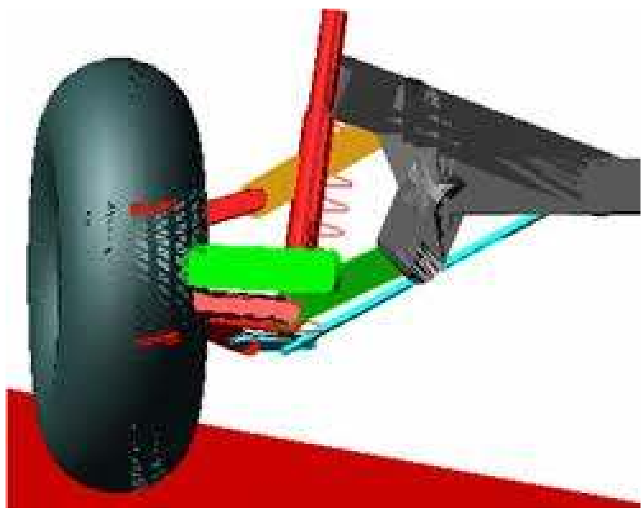


Figure 4.1 Tire suspension system (photo courtesy azom.com).

Modeling and simulation of vehicle suspension can help in predicting the performance of the system as well as inform the proper functioning before wasting time and money on producing it. An appropriate mathematical model used for modeling and simulation can always predict the system performance accurately. A vehicle suspension system is complex with multiple degrees of freedom. It’s main purpose being to isolate the vehicle from the road inputs. Various aspects of dynamics associated with vehicle put various requirements on the components of the suspension system. Luxurious cars tend towards passengers comfort while racing cars tends towards better road holdings.

For analyzing the vibration characteristics of the vehicle, equations of motion have to be formulated. Various models from a single degree of freedom model to a complex model having multiple degrees of freedom have been developed for studying the suspension system performance. However, the system is simplified by considering some dominating modes and modal approximations. For instance, a 'Quarter Car Model' has been extensively used to study the dynamic behavior of the vehicle suspension system. This is basically a linear lumped mass parameter model with two degrees of freedom. This model is used to obtain a qualitative insight into the performance of the suspension, in particular the effects of sprung mass and unsprung mass, stiffness of the suspension spring and tires, damping of the shock absorbers and tires on the vehicle vibration.

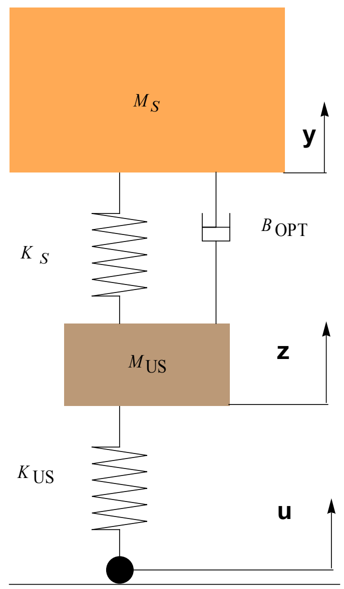


Figure 4.2 ‘Mathematica Model for Quarter Car Model’.

The 2-Degree-of-Freedom model shown in figure 2.1 includes an unsprung mass representing the wheels and associated components and a sprung mass representing the vehicle body. At a particular point of time, let y and z are the vertical displacements of the sprung and unsprung mass respectively due to the excitation from the rough road surface. The displacements y and z are measured from the static equilibrium positions so that we can neglect the gravity term while writing the equation of motion for the masses. This model is referred as a ‘quarter car model’. Usually the vehicle mass and the tire stiffness are the fixed parameters for a given model and other parameters are decided base on the frequency response characteristics. In the above model all the parameters are given fixed value except the vibrations which are generated by irregular road surface.

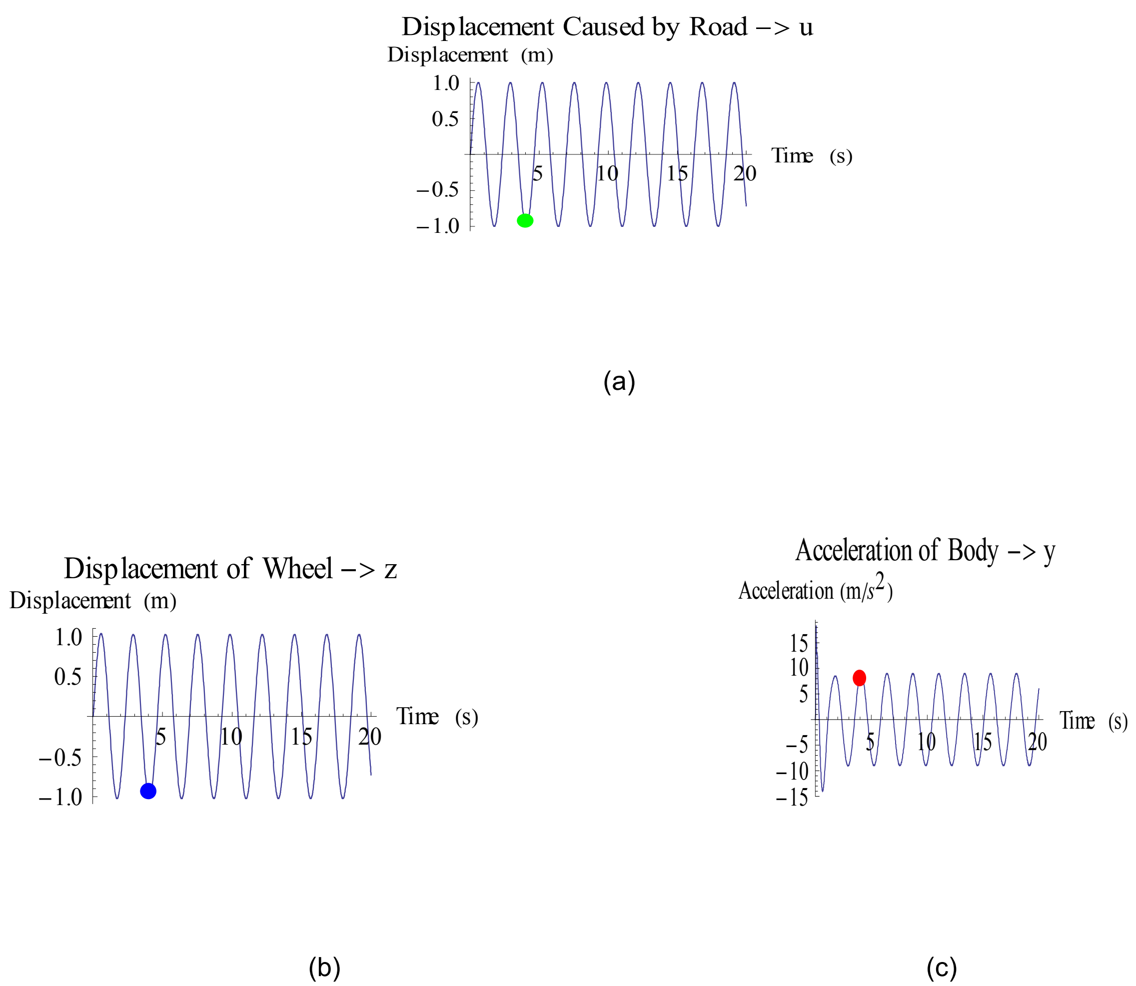


Figure 4.3 Graph. a) Displacement of road vs. time. b) Displacement of wheel vs. time. c) Acceleration of body vs. time

4.2 Vibration Absorber Model

Unlike vibration isolator which is a system of spring and damper between the source of vibration and the mass in consideration, vibration absorber is attachment of an auxiliary mass to a main mass with the help of spring. The reason for doing this is to suppress the vibrations to the main mass. The stiffness and the mass of the auxiliary mass are chosen so as to fulfill the above need. These systems are called as 'dynamic vibration absorbers' or 'tuned vibration absorbers'. These vibration absorbers are designed to work at a specific frequency and are less effective at other times [5]. They are mounted to reduce discomfort, damage or in some cases structure failures. Their applications include rotating machineries and pacifying the effects of earthquakes in tall buildings. One very famous example is the vibration absorber at Taipei 101.

Figure 4.4 Vibration absorber at Taipei 101 (photo courtesy theengineer.co.uk).

The tower could be subjected to earthquakes, typhoons and fierce winds--major challenges to the rigidity of the building. The remedy for these potential seismic and atmospheric assaults is this 730-ton tuned mass damper (TMD). It acts like a giant pendulum to counteract the building's movement--reducing sway due to wind by 30 to 40 percent. Constructed by specialty engineering firm Motioneering, the damper was too heavy to be lifted by crane and had to be assembled on-site. Eight steel cables form a sling to support the ball, while eight viscous dampers act like shock absorbers when the sphere shifts. Able to move 5 ft. in any direction, the Taipei TMD is the world's largest and heaviest. This gold-colored orb is on view from restaurants, bars and observation decks between the 88th and 92nd stories. A bumper ring prevents the ball from swaying too far, should that much swaying ever need to occur [7].

Another application of the absorber is in the aviation industry. They are used to suppress the vibrations in the wings of a plane during flight. The Mathematica model is shown below;

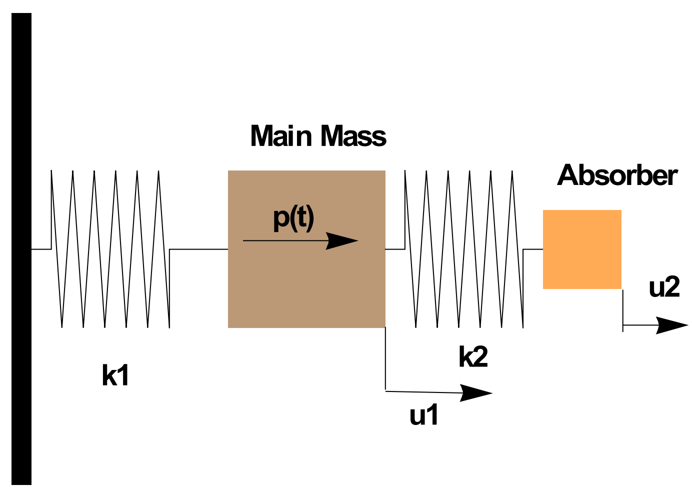


Figure 4.5 Vibration absorber system.

The equation of motion for the above system can be determined by similar technique as shown in the chapter 3 section 3.1. As all the other models this simple animation can demonstrate the effect of the harmonic force p (t) on the main mass. This load has an excitation frequency near the undamped natural frequency of the system this produces amplified vibrations. An auxiliary mass absorber is attached with a spring to reduce the effect of the harmonic load which is being applied to the main mass. The demonstration allows making changes in stiffness in the springs and the amplitude of the applied force. The mass of the absorber is related to that of main mass thus having no control on it. The amplitude and the frequency of the applied force can be edited according to the need. Some values like undamped natural frequency of system, angular natural frequency of absorber are displayed in the graphic box to attain required conditions. These conditions are to set excitation frequency near undamped natural frequency of system and set angular natural frequency of absorber equal to excitation frequency.

Hear in an output of the displacements of main mass and the absorber for a given condition;

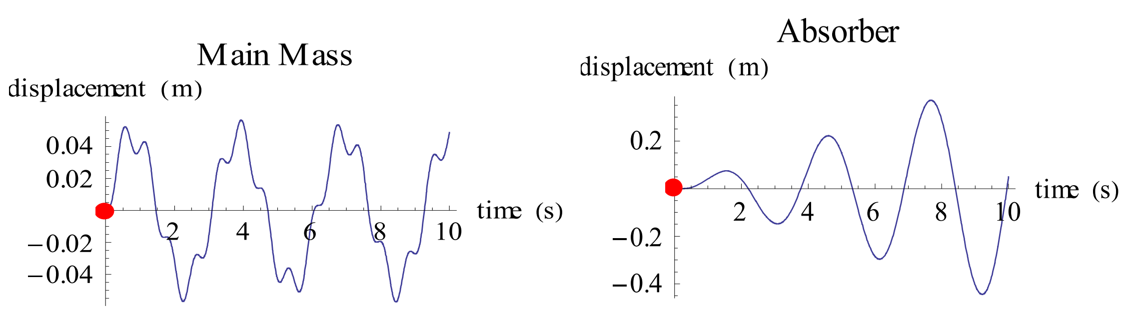


Figure 4.6 Response for tuned absorber.

As it can be seen from the figure 4.2.3 the displacement for the main mass is much less than that of the absorber. The vibration of the main mass is taken by the absorber. If this experiment is repeated with no or very low stiffness for the absorber spring all the vibrations will remain in the main mass.

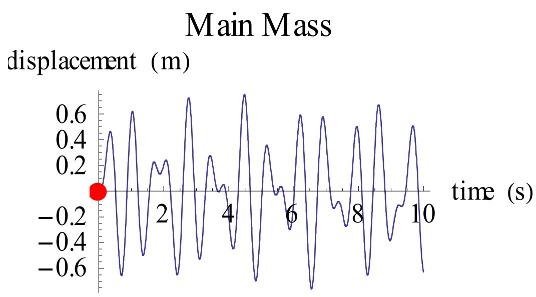


Figure 4.7 Response for untuned absorber.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Result

Here are small snapshots of a tutorial in CDF format:-

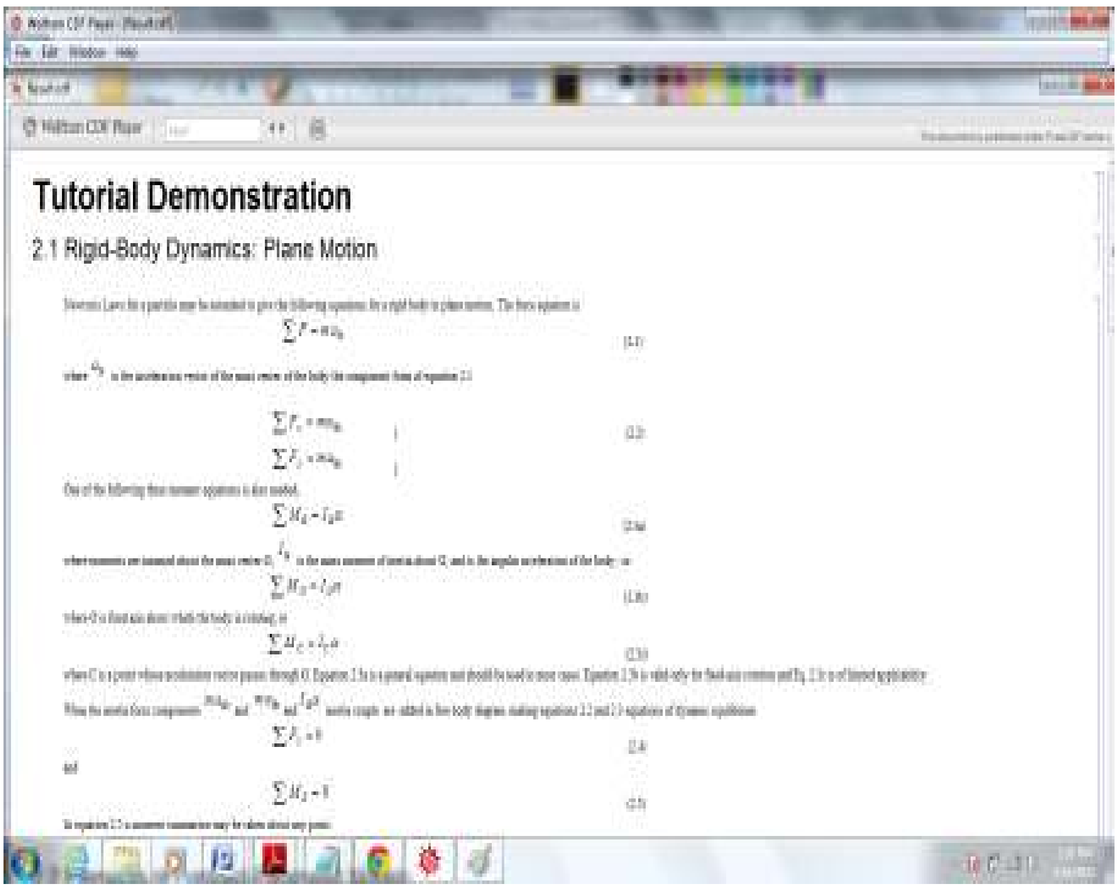


Figure 5.1 Snapshot 1

Figure 5.2 Snapshot 2

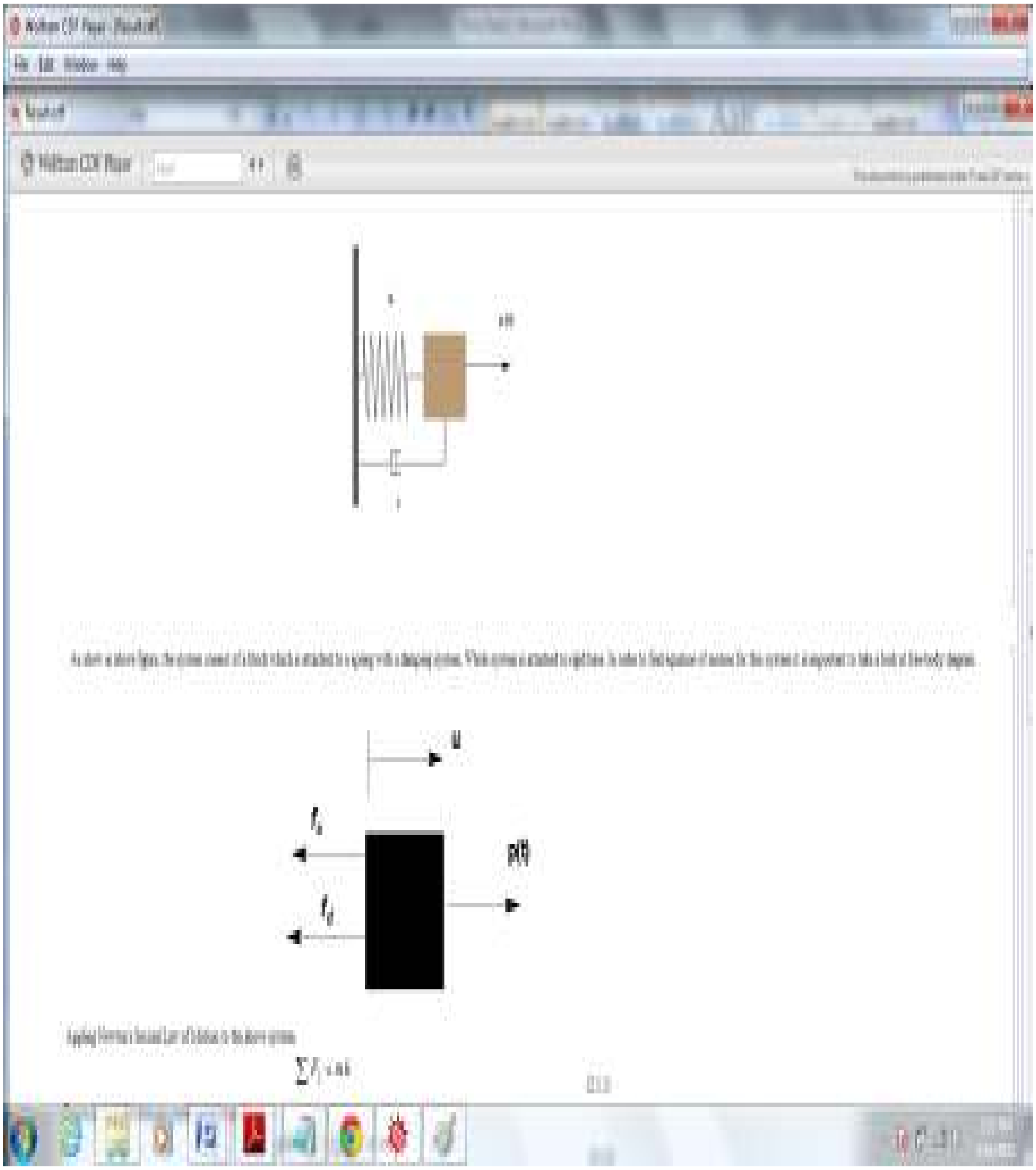


Figure 5.3 Snapshot 3

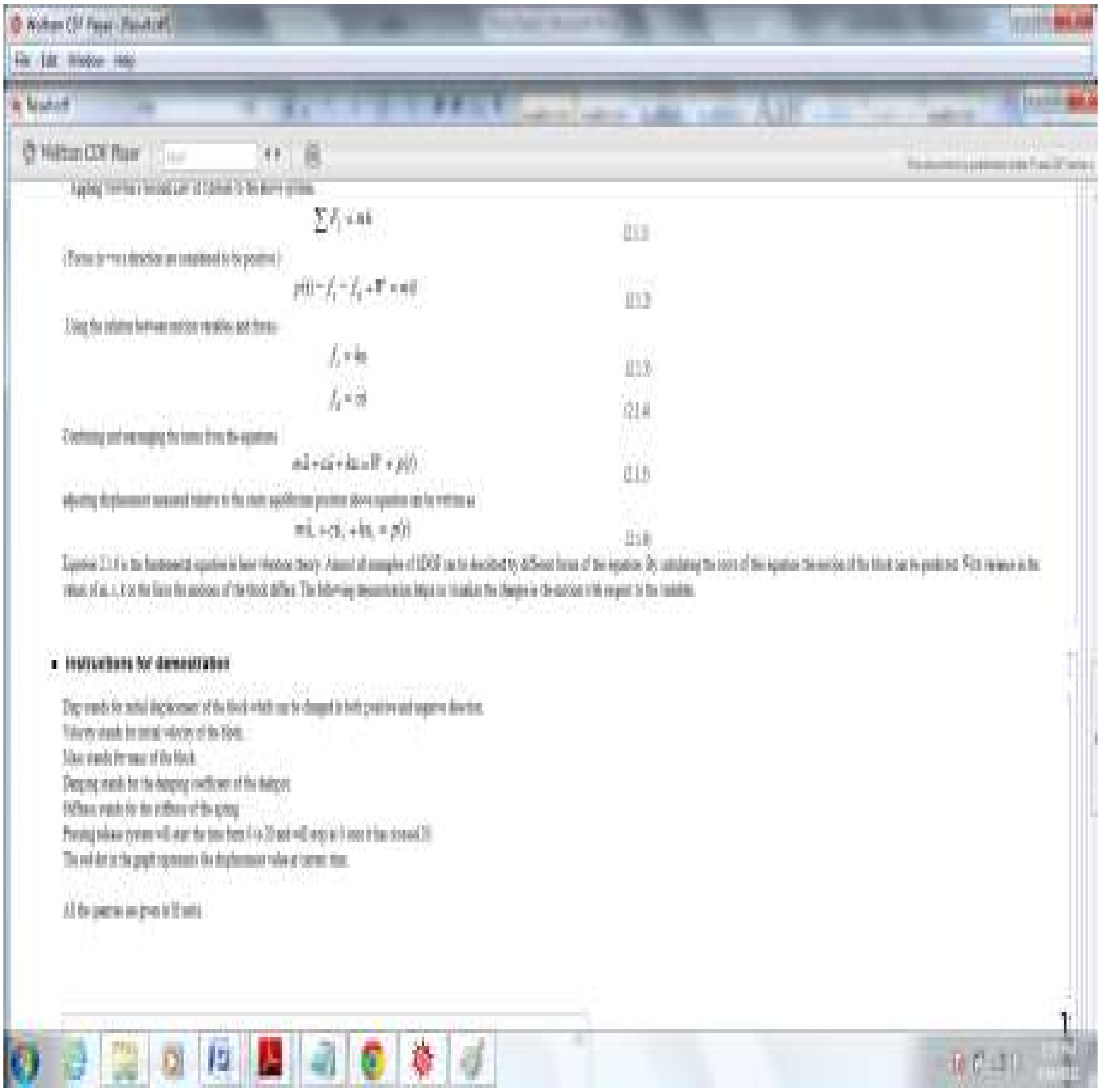


Figure 5.4 Snapshot 4

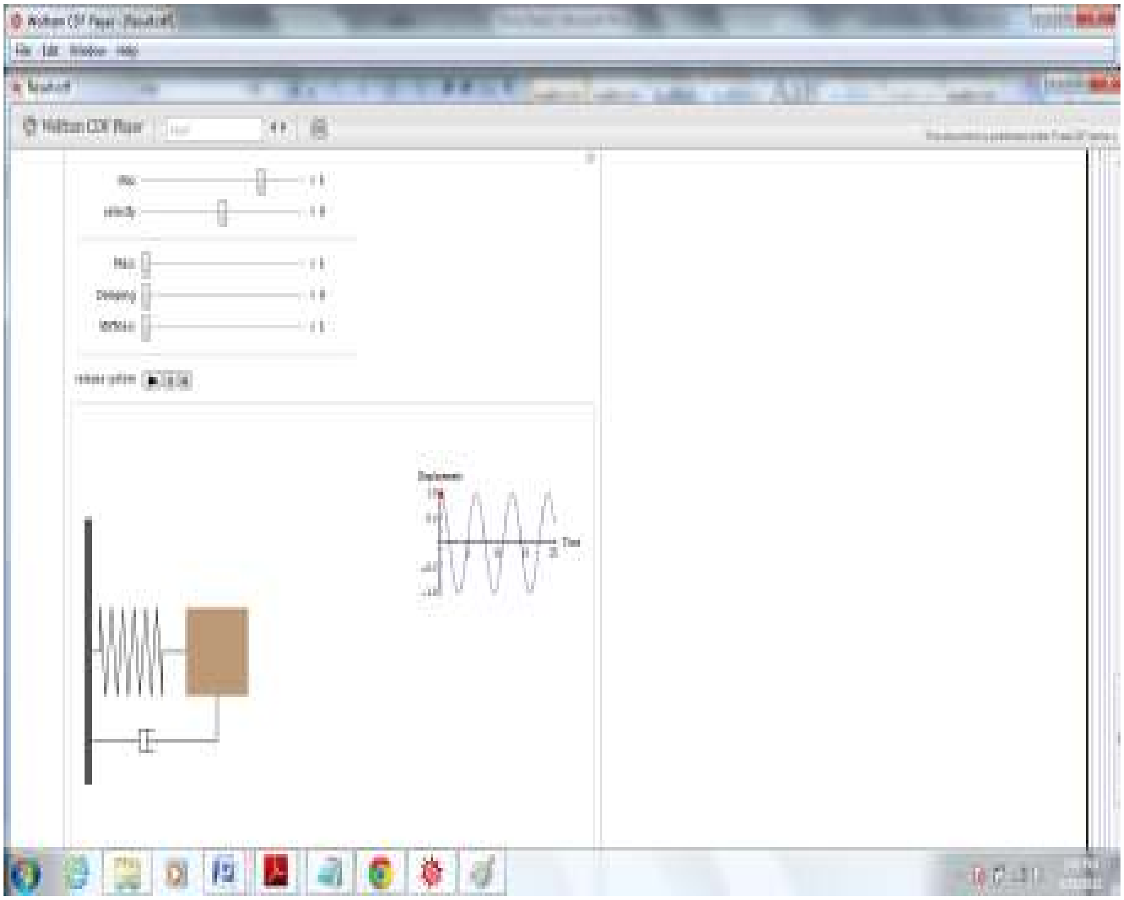


Figure 5.5 Snapshot 5

5.2 Limitations

The main limitation of using the CDF format is that it cannot be edited unless you have a proper version of Mathematica installed in your system. The limits of the parameters that can be varied in a CDF demonstration are always programmer defined.

5.3 Future Work

Several more demonstrations can be made with more innovations. They can also be applied to other fields in Mechanical Engineering. A three dimensional approach can be taken into consideration for making these demonstrations more realistic in nature.

APPENDIX A

MATHEMATICA INPUT CODES FOR THE DEMOSTRATIONS

**Nomenclature used in the codes does not correspond to one stated in the beginning.

<u>Spring Mass System</u>

```
Manipulate[If[run == 5, run = 0]; cd := 1.5*(2*Sqrt[st*ms]);
 w := N[Sqrt[st/(ms)]]; w1 := N[w/(2*\[Pi])]; Tn := N[1/w1];
 U := (disp + velocity)*Tn; nt := run/Tn; da := n*cd;
 sol = NDSolve[{ms*r''[t] + da*r'[t] + st*r[t] == 0, {r[0] == disp,
     r'[0] == velocity}}, {r[t], r'[t]}, {t, 0, 5},
   Method -> "StiffnessSwitching", InterpolationOrder -> All];
 sold[t_] = r[t] /. sol[[1, 1]];
 l = sold[run];

 pl = Plot[sold[t], {t, 0, 5}, ImageSize -> {300, 300},
   PlotRange -> Full, AxesLabel -> {"Time (s)", "Displacement (m)"},
   Epilog -> {PointSize[Large], Red, Point[{run, l}]}] ;


 gr = Graphics[{Line[{{-0.5, 0}, {0, 0}, {0, 2}}], {Lighter[Black],
     Rectangle[{-1, -6}, {-0.5, 6}]},
    Line[{{3 + l, -2}, {3 + l, 0}, {4.5 + l, 0}}], {Lighter [Brown],
     Rectangle[{4.5 + l, -2}, {8.5 + l, 2}]},
    Line[{{6.5 + l, -2}, {6.5 + l, -4}, {3, -4}}],
    Line[{{3, -3.5}, {3, -4.5}}],
    Line[{{3.5, -3.5}, {2.5, -3.5}, {2.5, -4.5}, {3.5, -4.5}}],
    Line[{{2.5, -4}, {-0.5, -4}}],
    Line[Table[{3 nx/11 + l*nx/11, 2 Cos[\[Pi]*nx]}, {nx, 0, 11}]],
    Text["\!\(\*SubscriptBox[\(f\), \(n\)]\)(Hz) =", {4, 3}],
    Text[w1, {6, 3}],
    Text["\!\(\*SubscriptBox[\(T\), \(n\)]\)(s) =", {4, 4}],
    Text[Tn, {6, 4}]}, PlotRange -> {{-1, 20}, {-6, 6}},
   ImageSize -> {400, 400}]; Grid[{{gr, pl}}, Alignment -> Top],
 {{disp, 0, "displacement (m)"}, -0.1, 0.1, .05,
  Appearance -> "Labeled"}, {{velocity, 0, "velocity (m/s)"}, -1,
  1, .05, Appearance -> "Labeled"}, Delimiter, {{ms, 5, "Mass (kg)"},
  5, 10, .05,
  Appearance -> "Labeled"}, {{n, 0, "% of Criticle damping"}, 0, 1,
  0.01, Appearance -> "Labeled"}, {{st, 200, "Stiffness (N/m)"}, 200,
  750, 1, Appearance -> "Labeled"}, Delimiter, {{run, 0,
   "release system"}, 0, 5, .01, ControlType -> Trigger,
  AnimationRate -> 1}, SynchronousUpdating -> False,
 SaveDefinitions -> True, AutorunSequencing -> {1, 2},
 ControlPlacement -> Left, TrackedSymbols -> True]
```

<u>Various Inputs for Spring Mass System</u>

<u>Step Force Input</u>

```
Manipulate[If[run == 10, run = 0]; p = a; w := N[Sqrt[st/ms]];
 w1 := N[w/(2*\[Pi])]; Tn := N[1/w1]; cd := 1.5*(2*Sqrt[st*ms]);
 da := n*cd; U = a/st;
 sol = NDSolve[{ms*r''[t] + da*r'[t] + st*r[t] == p, {r[0] == 0,
```

```
    r'[0] == 0}}, {r[t], r'[t]}, {t, 0, 10},
   Method -> "StiffnessSwitching", InterpolationOrder -> All];
 sold[t_] = r[t] /. sol[[1, 1]];
 l = sold[run]/U;
 plt1 = Plot[p, {t, 0, 10}, ImageSize -> {200, 200},
   AxesLabel -> {"Time (s)", "Load (N)"},
   Epilog -> {PointSize[Large], Red, Point[{run, p}]}];
 plt2 = Plot[sold[t]/U, {t, 0, 10}, ImageSize -> {200, 200},
   PlotRange -> Full, AxesLabel -> {"Time (s)", "Displacement"},
   Epilog -> {PointSize[Large], Blue, Point[{run, l}]}] ;
 grp = Graphics[{Line[{{-0.5, 0}, {0, 0}, {0, 2}}], {Brown,
     Rectangle[{-1, -6}, {-0.5, 6}]},
    Line[{{3 + l, -2}, {3 + l, 0}, {4.5 + l, 0}}], {{Lighter[Blue],
      Rectangle[{4.5 + l, -2}, {8.5 + l, 2}]}},
    Line[{{6.5 + l, -2}, {6.5 + l, -4}, {3, -4}}],
    Line[{{3, -3.5}, {3, -4.5}}],
    Line[{{3.5, -3.5}, {2.5, -3.5}, {2.5, -4.5}, {3.5, -4.5}}],
    Line[{{2.5, -4}, {-0.5, -4}}],
    Line[Table[{3 nx/11 + l*nx/11, 2 Cos[\[Pi]*nx]}, {nx, 0, 11}]],
    Text["\!\(\*SubscriptBox[\(f\), \(\(n\)\(\\\ \)\)]\)(Hz) =", {4,
      3}], Text[w1, {6, 3}],
    Text["\!\(\*SubscriptBox[\(T\), \(n\)]\)(S) =", {4, 4}],
    Text[Tn, {6, 4}]}, PlotRange -> {{-1, 20}, {-6, 6}},
   ImageSize -> {400, 400}];
 Grid[{{grp, plt1}, {SpanFromAbove, plt2}}, Alignment -> Top],
 {{ms, 5, "Mass (kg)"}, 5, 5,
  Appearance -> "Labeled"}, {{n, 0, "% of Criticle damping"}, 0, 1,
  0.01, Appearance -> "Labeled"}, {{st, 200, "Stiffness (N/m)"}, 200,
  750, .05,
  Appearance -> "Labeled"}, Delimiter, {{a, 1, "Input Magnitude(N)"},
  1, 10, .05,
  Appearance -> "Labeled"}, Delimiter, {{run, 0, "release system"}, 0,
   10, .01, ControlType -> Trigger, AnimationRate -> 1},
 SaveDefinitions -> True, SynchronousUpdating -> False,
 AutorunSequencing -> {1, 2}, ControlPlacement -> Left,
 TrackedSymbols -> True]
```

<u>Rectangular Pulse Force Input</u>

```
Manipulate[If[run == 10, run = 0]; w := N[Sqrt[st/ms]];
 w1 := N[w/(2*\[Pi])]; Tn := N[1/w1]; f := 5*Tn; U = 10/st;
 c[t_] := If[t < td, 10, 0]; cd := 1.5*(2*Sqrt[st*ms]); da := n*cd;
 sol = NDSolve[{ms*r''[t] + da*r'[t] + st*r[t] == c[t], {r[0] == 0,
     r'[0] == 0}}, {r[t], r'[t]}, {t, 0, 10},
   Method -> "StiffnessSwitching", InterpolationOrder -> All];
 sold[t_] = r[t] /. sol[[1, 1]];
 l = sold[run]/U;
 plt1 = Plot[c[t], {t, 0, 20}, ImageSize -> {200, 200},
   AxesLabel -> {"Time (s)", "Load (N)"},
   Epilog -> {PointSize[Large], Red, Point[{run, c[run]}]}];
 plt2 = Plot[{sold[t]/U}, {t, 0, 10}, ImageSize -> {200, 200},
```

```
  PlotRange -> Full, AxesLabel -> {"Time (s)", "Displacement "},
  Epilog -> {PointSize[Large], Blue, Point[{run, l}]}] ;
 grp = Graphics[{Line[{{-0.5, 0}, {0, 0}, {0, 2}}], {Brown,
    Rectangle[{-1, -6}, {-0.5, 6}]},
   Line[{{3 + l, -2}, {3 + l, 0}, {4.5 + l, 0}}], {{Lighter[Blue],
     Rectangle[{4.5 + l, -2}, {8.5 + l, 2}]}},
   Line[{{6.5 + l, -2}, {6.5 + l, -4}, {3, -4}}],
   Line[{{3, -3.5}, {3, -4.5}}],
   Line[{{3.5, -3.5}, {2.5, -3.5}, {2.5, -4.5}, {3.5, -4.5}}],
   Line[{{2.5, -4}, {-0.5, -4}}],
   Line[Table[{3 nx/11 + l*nx/11, 2 Cos[\[Pi]*nx]}, {nx, 0, 11}]],
   Text["\!\(\*SubscriptBox[\(f\), \(n\)]\)(Hz) =", {4, 3}],
   Text[w1, {6, 3}],
   Text["\!\(\*SubscriptBox[\(T\), \(n\)]\)(s) =", {4, 4}],
   Text[Tn, {6, 4}], Text["Adviced Max pulse Duration =", {3, 5}],
   Text[f, {8, 5}]}, PlotRange -> {{-1, 20}, {-6, 8}},
  ImageSize -> {400, 400}];
 Grid[{{grp, plt1}, {SpanFromAbove, plt2}}, Alignment -> Top],
 {{ms, 5, "Mass(kg)"}, 5, 5,
  Appearance -> "Labeled"}, {{n, 0, "% of Criticle damping"}, 0, 1,
  0.01, Appearance -> "Labeled"}, {{st, 200, "Stiffness (N/m)"}, 200,
  750, 1, Appearance -> "Labeled"}, Delimiter, {{td, 0,
   "Pluse Duration (s)"}, 0, 5, .05,
  Appearance -> "Labeled"}, Delimiter, {{run, 0, "release system"}, 0,
   10, .01, ControlType -> Trigger, AnimationRate -> 1},
 SaveDefinitions -> True, SynchronousUpdating -> False,
 AutorunSequencing -> {1, 2}, ControlPlacement -> Left,
 TrackedSymbols -> True]
```

<u>Ramp Force Input</u>

```
Manipulate[If[run == 10, run = 0]; p = 1; w := Sqrt[st/ms];
 w1 := N[w/(2*\[Pi])]; Tn := N[1/w1]; cd := 1.5*(2*Sqrt[st*ms]);
 da := n*cd; f := 5*Tn; c[t_] := If[t < td, t, td];
 sol = NDSolve[{ms*r''[t] + da*r'[t] + st*r[t] == c[t], {r[0] == 0,
     r'[0] == 0}}, {r[t], r'[t]}, {t, 0, 10},
   Method -> "StiffnessSwitching", InterpolationOrder -> All];
 U := td/st;
 sold[t_] = r[t] /. sol[[1, 1]];
 l = sold[run]/U;
 plt1 = Plot[c[t], {t, 0, 10}, ImageSize -> {200, 200},
   AxesLabel -> {"Time (s)", "Load (N)"},
   Epilog -> {PointSize[Large], Red, Point[{run, c[run]}]}];
 plt2 = Plot[sold[t]/U, {t, 0, 10}, ImageSize -> {200, 200},
   PlotRange -> Full, AxesLabel -> {"Time (s)", "Displacement"},
   Epilog -> {PointSize[Large], Blue, Point[{run, l}]}] ;
 grp = Graphics[{Line[{{-0.5, 0}, {0, 0}, {0, 2}}], {Brown,
    Rectangle[{-1, -6}, {-0.5, 6}]},
   Line[{{3 + l, -2}, {3 + l, 0}, {4.5 + l, 0}}], {{Lighter[Blue],
     Rectangle[{4.5 + l, -2}, {8.5 + l, 2}]}},
   Line[{{6.5 + l, -2}, {6.5 + l, -4}, {3, -4}}],
```

```
   Line[{{3, -3.5}, {3, -4.5}}],
   Line[{{3.5, -3.5}, {2.5, -3.5}, {2.5, -4.5}, {3.5, -4.5}}],
   Line[{{2.5, -4}, {-0.5, -4}}],
   Line[Table[{3 nx/11 + l*nx/11, 2 Cos[\[Pi]*nx]}, {nx, 0, 11}]],
   Text["\!\(\*SubscriptBox[\(f\), \(n\)]\) =", {4, 3}],
   Text[w1, {6, 3}],
   Text["\!\(\*SubscriptBox[\(T\), \(n\)]\) =", {4, 4}],
   Text[Tn, {6, 4}], Text["Advised Max pulse Duration =", {3, 5}],
   Text[f, {8, 5}]}, PlotRange -> {{-1, 20}, {-6, 6}},
  ImageSize -> {400, 400}];
 Grid[{{grp, plt1}, {SpanFromAbove, plt2}}, Alignment -> Top],
 {{ms, 5, "Mass (kg)"}, 5, 5,
  Appearance -> "Labeled"}, {{n, 0, "% of Criticle damping"}, 0, 1,
  0.01, Appearance -> "Labeled"}, {{st, 200, "Stiffness (N/m)"}, 200,
  750, .05,
  Appearance -> "Labeled"}, Delimiter, {{td, 0.1,
   "Pulse Duration (s)"}, 0.1, 5, .05,
  Appearance -> "Labeled"}, Delimiter, {{run, 0, "release system"}, 0,
   10, .01, ControlType -> Trigger, AnimationRate -> 1},
 SaveDefinitions -> True, SynchronousUpdating -> False,
 AutorunSequencing -> {1, 2}, ControlPlacement -> Left,
 TrackedSymbols -> True]
```

<u>Triangular Pulse Force Input</u>

```
Manipulate[If[run == 10, run = 0]; c[t_] := If[t <= td, td - t, 0];
 w := Sqrt[st/ms]; w1 := N[w/(2*\[Pi])]; Tn := N[1/w1];
 cd := 1.5*(2*Sqrt[st*ms]); da := n*cd; f := 5*Tn; U := td/st;
 sol = NDSolve[{ms*r''[t] + da*r'[t] + st*r[t] == c[t], {r[0] == 0,
    r'[0] == 0}}, {r[t], r'[t]}, {t, 0, 10}];
 sold[t_] = r[t] /. sol[[1, 1]];
 l = sold[run]/U;
 plt1 = Plot[c[t], {t, 0, 10}, ImageSize -> {200, 200},
   PlotRange -> Full, AxesLabel -> {"Time (s)", "Load (N)"},
   Epilog -> {PointSize[Large], Red, Point[{run, c[run]}]}];
 plt2 = Plot[sold[t]/U, {t, 0, 5}, ImageSize -> {200, 200},
   PlotRange -> Full, AxesLabel -> {"Time (s)", "Displacement"},
   Epilog -> {PointSize[Large], Blue, Point[{run, l}]}] ;
 grp = Graphics[{Line[{{-0.5, 0}, {0, 0}, {0, 2}}], {Brown,
    Rectangle[{-1, -6}, {-0.5, 6}]},
   Line[{{3 + l, -2}, {3 + l, 0}, {4.5 + l, 0}}], {{Lighter[Blue],
     Rectangle[{4.5 + l, -2}, {8.5 + l, 2}]}},
   Line[{{6.5 + l, -2}, {6.5 + l, -4}, {3, -4}}],
   Line[{{3, -3.5}, {3, -4.5}}],
   Line[{{3.5, -3.5}, {2.5, -3.5}, {2.5, -4.5}, {3.5, -4.5}}],
   Line[{{2.5, -4}, {-0.5, -4}}],
   Line[Table[{3 nx/11 + l*nx/11, 2 Cos[\[Pi]*nx]}, {nx, 0, 11}]],
   Text["\!\(\*SubscriptBox[\(f\), \(n\)]\) =", {4, 3}],
   Text[w1, {6, 3}],
   Text["\!\(\*SubscriptBox[\(T\), \(n\)]\) =", {4, 4}],
```

```
   Text[Tn, {6, 4}], Text["Advised Max pulse Duration =", {3, 5}],
   Text[f, {8, 5}]}, PlotRange -> {{-1, 20}, {-6, 6}},
  ImageSize -> {400, 400}];
Grid[{{grp, plt1}, {SpanFromAbove, plt2}}, Alignment -> Top],
{{ms, 5, "Mass (kg)"}, 5, 5,
 Appearance -> "Labeled"}, {{n, 0, "% of Criticle damping"}, 0, 1,
 0.01, Appearance -> "Labeled"}, {{st, 200, "Stiffness (N/m)"}, 200,
 750, .05,
 Appearance -> "Labeled"}, Delimiter, {{td, 0.1,
  "Pulse Duration (s)"}, 0.1, 5, .05,
 Appearance -> "Labeled"}, Delimiter, {{run, 0, "release system"}, 0,
  10, .01, ControlType -> Trigger, AnimationRate -> 1},
SaveDefinitions -> True, SynchronousUpdating -> False,
AutorunSequencing -> {1, 2}, ControlPlacement -> Left,
TrackedSymbols -> True]
```

Moving Base for Spring Mass System

```
Manipulate[If[run == 10, run = 0]; z[t_] := Cos[\[CapitalOmega] t];
 cd := 1.5*(2*Sqrt[st*ms]); da := n*cd; w := N[Sqrt[st/(ms)]];
 rt = 0.5; \[CapitalOmega] := rt*w;
 z'[t] == D[z[t], t];
 sol = NDSolve[{ms*r''[t] + da*r'[t] + st*r[t] ==
     da*z'[t] + st*z[t], {r[0] == disp, r'[0] == 0}}, {r[t],
    r'[t]}, {t, 0, 10}, Method -> "StiffnessSwitching",
   InterpolationOrder -> All];
 sold[t_] = r[t] /. sol[[1, 1]];
 l = sold[run]; f = z[run];
 pt = Plot[sold[t], {t, 0, 10}, ImageSize -> {300, 300},
   AxesLabel -> {"Time (s)", "Block Displacement (m)"},
   PlotRange -> All,
   Epilog -> {PointSize[Large], Red, Point[{run, (l)}]}];
 gr = Graphics[{Line[{{-0.5 + f, 0}, {f, 0}, {f, 2}}], {Lighter [
      Black], Rectangle[{-1 + f, -6}, {-0.5 + f, 6}]},
    Line[{{3 + l + f, -2}, {3 + l + f, 0}, {4.5 + l + f,
       0}}], {Lighter [Orange],
     Rectangle[{4.5 + l + f, -2}, {8.5 + l + f, 2}]},
    Line[{{6.5 + l + f, -2}, {6.5 + l + f, -4}, {3 + f, -4}}],
    Line[{{3 + f, -3.5}, {3 + f, -4.5}}],
    Line[{{3.5 + f, -3.5}, {2.5 + f, -3.5}, {2.5 + f, -4.5}, {3.5 +
        f, -4.5}}], Line[{{2.5 + f, -4}, {-0.5 + f, -4}}],
    Line[Table[{3 nx/11 + l*nx/11 + f, 2 Cos[\[Pi]*nx]}, {nx, 0,
       11}]], Text[
     "\!\(\*SubscriptBox[\(\[Omega]\), \(n\)]\)(rad/s) =", {4, 3}],
    Text[w, {7, 3}], Text["Forcing fr(rad/s) =", {4, 4}],
    Text[\[CapitalOmega], {8, 4}]}, PlotRange -> {{-3, 20}, {-6, 6}},
   ImageSize -> {400, 400}];
 Grid[{{gr, pt}},
  Alignment -> Top], {{disp, 0.1, "initial displacement (m)"}, -0.1,
  0.1, .05,
  Appearance -> "Labeled"}, Delimiter, {{ms, 1, "Mass (kg)"}, 1,
```

```
 5, .05, Appearance -> "Labeled"}, {{n, 0, "% of Criticle damping"},
 0, 1, 0.01, Appearance -> "Labeled"}, {{st, 0, "Stiffness (N/m)"},
 0, 750, 1,
 Appearance -> "Labeled"}, Delimiter, {{run, 0, "release system"}, 0,
  10, .01, ControlType -> Trigger, AnimationRate -> 0.8},
SaveDefinitions -> True, SynchronousUpdating -> False,
AutorunSequencing -> {1, 2}, TrackedSymbols -> True]
```

Non Linear Pendulum

```
Manipulate[
 If[run == 10, run = 0];
 Module[{sol, \[Theta], t},
  sol = NDSolve[{\[Theta]''[t] + (9.8/l)*Sin[\[Theta][t]] ==
      0, \[Theta][0] == theta, \[Theta]'[0] ==
      thetaPrime}, \[Theta], {t, 0, 10},
    Method -> "StiffnessSwitching"];
  With[{angle = \[Theta][run] /. sol[[1, 1]]},
   Plot[\[Theta][t] /. sol[[1, 1]], {t, 0, 10},
    AxesLabel -> {"Time (s)", "Displacement (rad)"},
    PlotRange -> {{0, 10}, {-2.5, 2.5}}, AspectRatio -> 1,
    ImageSize -> {300, 200}]
    Graphics[
    {Green, Rectangle[{-3, 0}, {3, .5}], Brown,
     Line[{{0, -1}, {(l)* Sin[angle], -((l)*Cos[angle])}}], Black,
     Disk[{0, 0}, {0.5, 1}, {180 Degree, 360 Degree}], Red,
     Disk[{l *Sin[angle], -(l*Cos[angle])}, .3]},
    PlotRange -> {{-7, 7}, {.5, -5.5}}, ImageSize -> {500, 300}
    ]
   ]
  ],
 {{l, 3, "Length (m)"}, 1.5, 5, .1, Appearance -> "Labeled"},
 {{theta, \[Pi]/4, "Displacement (rad)"}, 0, \[Pi]/3, 0.01,
  Appearance -> "Labeled"},
 {{thetaPrime, 0, "Velocity(m/s)"}, 0, 2.5, 0.5,
  Appearance -> "Labeled"},
 {{run, 0, "release system"}, 0, 10, .001, ControlType -> Trigger,
  AnimationRate -> 1}, SynchronousUpdating -> False,
 AutorunSequencing -> {1, 2}, SaveDefinitions -> True
 ]
```

Free Transverse Vibration Of Bernoulli-Euler Beams

Simply Supported Beam

```
Manipulate[If[t == 1, t = 0];
 e = 3*10^7; i = 170; a = 8.84; \[Rho] = 7.86*10^3;
 g = ((e*i)/(\[Rho]*a))^0.5; \[Omega]r = ((r*Pi)/L)*g;
 L = 15; p5 = {{14.5, 0.5}, {15, 1}, {15.5, 0.5}};
```

```
p4 = {{-0.5, 0.5}, {0, 1}, {0.5, 0.5}};
Graphics[{Polygon[{p4, p5}],

  Plot[1 + Sin[r*Pi*x/L]*Sin[\[Omega]r t], {x, 0, L}, Axes -> False,
    PlotStyle -> {Red, Thick}, PlotRange -> {{-2, 4}, {-4, 4}},
    ImageSize -> {200, 200}][[1]]}, PlotRange -> {{-1, 16}, {-4, 4}},
  ImageSize -> {500, 500}

 ],
{{r, 1, "Mode Number"}, {1, 2, 3, 4, 5}, ControlType -> Setter},


Delimiter, {{t, 0, "Vibrate"}, 0, 1, .1, ControlType -> Trigger,
 AnimationRate -> 0.1}, SaveDefinitions -> True,
SynchronousUpdating -> False, AutorunSequencing -> {1, 2},
ControlPlacement -> Left, TrackedSymbols -> True

]
```

Cantilever Beam

```
Manipulate[If[t == 1, t = 0];

 a = 1.8751; b = 4.6941; c = 7.8548; d = 10.996; e = 3*10^7; i = 170;
 f = 8.84; \[Rho] = 7.86*10^3;
 g = ((e*i)/(\[Rho]*f))^0.5; \[Omega]n = r^2/l^2*g;
 k = (Cosh[r] + Cos[r])/(Sinh[r] + Sin[r]); l = 15;
 Graphics[{
   {Lighter [Blue], Rectangle [{-0.25, -0.5}, {0, 0.25}]},
   Plot[{{(Cosh[r/l*x] - Cos[r/l*x]) -
        k*(Sinh[r/l*x] - Sin[r/l*x])}*0.40*Sin[\[Omega]n t]}, {x, 0,
     l}, Axes -> False, PlotStyle -> {Red, Thick},
    PlotRange -> {{-2, 4}, {-4, 4}}][[1]]},
  PlotRange -> {{-2, 14}, {-4, 4}}, ImageSize -> {300, 300}
  ],

 {{r, a, "Mode Number"}, {a -> "Mode1", b -> "Mode2", c -> "Mode3",
   d -> "Mode4"}, ControlType -> Setter},


 Delimiter, {{t, 0, "Vibrate"}, 0, 1, .1, ControlType -> Trigger,
  AnimationRate -> 0.1}, SaveDefinitions -> True,
 SynchronousUpdating -> False, AutorunSequencing -> {1, 2},
 ControlPlacement -> Left, TrackedSymbols -> True]
```

Clamped and Clamped Beams

```
Manipulate[If[t == 1, t = 0];

 a = 4.7300; b = 7.8532; c = 10.9956; d = 14.1371; e = 3*10^7;
 i = 170; f = 8.84; \[Rho] = 7.86*10^3;
```

```
g = ((e*i)/(\[Rho]*f))^0.5; \[Omega]n = r^2/l^2*g;
k = (Cosh[r] - Cos[r])/(Sinh[r] - Sin[r]); l = 15;
Graphics[{
  {Lighter [Blue],
   Rectangle [{-0.125, -0.25}, {0, 0.25}]}, {Lighter [Blue],
   Rectangle [{15, -0.25}, {15.125, 0.25}]},
  Plot[{{(Cosh[r/l*x] - Cos[r/l*x]) -
       k*(Sinh[r/l*x] - Sin[r/l*x])}*0.40*Sin[\[Omega]n t]}, {x, 0,
     l}, Axes -> False, PlotStyle -> {Red, Thick},
    PlotRange -> {{-2, 8}, {-4, 4}}][[1]]},
 PlotRange -> {{-2, 16}, {-4, 4}}, ImageSize -> {300, 300}
 ],

{{r, a, "Mode Number"}, {a -> "Mode1", b -> "Mode2", c -> "Mode3",
  d -> "Mode4"}, ControlType -> Setter},


Delimiter, {{t, 0, "Vibrate"}, 0, 1, .1, ControlType -> Trigger,
 AnimationRate -> 0.1}, SaveDefinitions -> True,
SynchronousUpdating -> False, AutorunSequencing -> {1, 2},
ControlPlacement -> Left, TrackedSymbols -> True]
```

<u>Clamped and Supported</u>

```
Manipulate[If[t == 1, t = 0];

a = 3.9266; b = 7.06858; c = 10.2102; d = 13.3518; e = 3*10^7;
i = 170; f = 8.84; \[Rho] = 7.86*10^3;
g = ((e*i)/(\[Rho]*f))^0.5; \[Omega]n = r^2/l^2*g;
k = (Cosh[r] - Cos[r])/(Sinh[r] - Sin[r]); l = 15;
Graphics[{
  {Lighter [Blue], Rectangle [{-0.25, -0.5}, {0, 0.25}]},
  Polygon[{{14.5, -0.6}, {15, 0}, {15.5, -0.6}}],
  Plot[{{(Cosh[r/l*x] - Cos[r/l*x]) -
       k*(Sinh[r/l*x] - Sin[r/l*x])}*0.40*Sin[\[Omega]n t]}, {x, 0,
     l}, Axes -> False, PlotStyle -> {Red, Thick},
    PlotRange -> {{-2, 4}, {-4, 4}}][[1]]},
 PlotRange -> {{-2, 16}, {-4, 4}}, ImageSize -> {300, 300}
 ],

{{r, a, "Mode Number"}, {a -> "Mode1", b -> "Mode2", c -> "Mode3",
  d -> "Mode4"}, ControlType -> Setter},


Delimiter, {{t, 0, "Vibrate"}, 0, 1, .1, ControlType -> Trigger,
 AnimationRate -> 0.1}, SaveDefinitions -> True,
SynchronousUpdating -> False, AutorunSequencing -> {1, 2},
ControlPlacement -> Left, TrackedSymbols -> True]
```

Free- Free Beam

```
Manipulate[If[t == 1, t = 0];

 a = 4.7300; b = 7.8532; c = 10.9956; d = 14.1371; e = 3*10^7;
 i = 170; f = 8.84; \[Rho] = 7.86*10^3;
 g = ((e*i)/(\[Rho]*f))^0.5; \[Omega]n = r^2/l^2*g;
 k = (Cosh[r] - Cos[r])/(Sinh[r] - Sin[r]); l = 15;
 Graphics[{

   Plot[{{(Cosh[r/l*x] + Cos[r/l*x]) -
        k*(Sinh[r/l*x] + Sin[r/l*x])}*0.40*Sin[\[Omega]n t]}, {x, 0,
      l}, Axes -> False, PlotStyle -> {Red, Thick},
     PlotRange -> {{-2, 4}, {-4, 4}}][[1]]},
  PlotRange -> {{-2, 15}, {-4, 4}}, ImageSize -> {300, 300}
  ],

 {{r, a, "Mode Number"}, {a -> "Mode1", b -> "Mode2", c -> "Mode3",
   d -> "Mode4"}, ControlType -> Setter},


 Delimiter, {{t, 0, "Vibrate"}, 0, 1, .1, ControlType -> Trigger,
  AnimationRate -> 0.1}, SaveDefinitions -> True,
 SynchronousUpdating -> False, AutorunSequencing -> {1, 2},
 ControlPlacement -> Left, TrackedSymbols -> True]
```

Spring Mass Two Degrees of Freedom System

```
Manipulate[If[run == 10, run = 0];
 sol = NDSolve[{{m1*r1''[t] + r1'[t]*(c1 + c2) -
       r2'[t]*c2 + (k1 + k2)*r1[t] - k2*r2[t] == 0,
     m2*r2''[t] + c2*r2'[t] - c1*r1'[t] - k2*r1[t] + k2*r2[t] ==
      0}, {r1[0] == disp1, r2[0] == disp2, r1'[0] == 0,
     r2'[0] == 0}}, {r1[t], r2[t], r1'[t], r2'[t]}, {t, 0, 10},
   Method ->  "StiffnessSwitching", InterpolationOrder -> All];
 sold1[t_] = r1[t] /. sol[[1, 1]];
 sold2[t_] = r2[t] /. sol[[1, 2]];
 l = sold1[run];
 m = sold2[run];
 pl1 = Plot[{sold1[t], sold2[t]}, {t, 0, 10}, ImageSize -> {300, 300},
    PlotLabel -> "Block 1 & Block 2", PlotStyle -> {Blue, Red},
   PlotRange -> All, AxesLabel -> {"time (s)", "displacement (m)"},
   Epilog -> {{PointSize[Large], Blue, Point[{run, m}]},
     PointSize[Large], Red, Point[{run, l}]}];



 gr1 = Graphics[{Line[{{-0.5, 0}, {0, 0}, {0, 2}}],
    Rectangle[{-1, -6}, {-0.5, 6}],
```

```
   Line[{{3 + l, -2}, {3 + l, 0}, {4.5 + l, 0}}], {Lighter[Brown],
    Rectangle[{4.5 + l, -2}, {8.5 + l, 2}]},
   Line[Table[{3 nx/11 + l*nx/11, 2 Cos[\[Pi]*nx]}, {nx, 0, 11}]],
   Line[{{8.5 + l, 0}, {9 + l, 0}, {9 + l, 2}}],
   Line[{{-0.5, -3}, {2.5, -3}}],
   Line[{{3.5, -2.5}, {2.5, -2.5}, {2.5, -3.5}, {3.5, -3.5}}],
   Line[{{3, -2.5}, {3, -3.5}}],
   Line[{{3, -3}, {5.5 + l, -3}, {5.5 + l, -2}}],

   Line[Table[{9 + l + 3 nx/11 + m*nx/11, 2 Cos[\[Pi]*nx]}, {nx, 0,
      11}]], Line[{{12 + m + l, -2}, {12 + m + l, 0}, {13.5 + m + l,
      0}}], {Lighter[Orange],
    Rectangle[{13.5 + m + l, -2}, {17.5 + m + l, 2}]},
   Line[{{7.5 + l, -2}, {7.5 + l, -3}, {10.75 + l, -3}}],
   Line[{{11.75 + l, -2.5}, {10.75 + l, -2.5}, {10.75 +
       l, -3.5}, {11.75 + l, -3.5}}],
   Line[{{11.25 + l, -2.5}, {11.25 + l, -3.5}}],
   Line[{{11.25 + l, -3}, {14.5 + l + m, -3}, {14.5 + l + m, -2}}]

   }, PlotRange -> {{-1, 30}, {-6, 6}}, ImageSize -> {400, 400}];
 Grid[{{gr1, pl1}},
  Alignment -> Top], {{disp1, 0, "Initial Disp 1 (m)"}, -0.1,
  0.1, .05,
  Appearance -> "Labeled"}, {{disp2, 0, "Initial Disp 2 (m)"}, -0.1,
  0.1, .05,
  Appearance -> "Labeled"}, Delimiter, {{m1, 5, "Mass1 (kg)"}, 5,
  10, .05, Appearance -> "Labeled"}, {{m2, 5, "Mass2 (kg)"}, 5,
  10, .05, Appearance -> "Labeled"}, {{k1, 200, "Stiffness 1 (N/m)"},
  200, 750, .05,
  Appearance -> "Labeled"}, {{k2, 200, "Stiffness 2 (N/m)"}, 200,
  750, .05,
  Appearance -> "Labeled"}, Delimiter, {{c1, 0, "Damping 1"}, 0,
  5, .05, Appearance -> "Labeled"}, {{c2, 0, "Damping 2"}, 0, 5, .05,
  Appearance -> "Labeled"}, Delimiter, {{run, 0, "release system"}, 0,
   10, .01, ControlType -> Trigger, AnimationRate -> 1},
 SaveDefinitions -> True, SynchronousUpdating -> False,
 AutorunSequencing -> {1, 2}, ControlPlacement -> Left,
 TrackedSymbols -> True]
```

<u>Spring Mass Three Degrees of Freedom System</u>

```
Manipulate[If[run == 10, run = 0];
 sol = NDSolve[{{m1*u1''[t] + (k1 + k2)*u1[t] - k2*u2[t] == 0,
     m2*u2''[t] - k2*u1[t] + (k2 + k3)*u2[t] - k3*u3[t] == 0,
     m3*u3''[t] - k3*u2[t] + k3*u3[t] == 0}, {u1[0] == disp1,
     u2[0] == disp2, u3[0] == disp3, u1'[0] == 0, u2'[0] == 0,
     u3'[0] == 0}}, {u1[t], u2[t], u3[t], u1'[t], u2'[t], u3'[t]}, {t,
     0, 10}, Method -> "StiffnessSwitching"];
 sold1[t_] = u1[t] /. sol[[1, 1]];
```

```
sold2[t_] = u2[t] /. sol[[1, 2]];
sold2[t_] = u3[t] /. sol[[1, 3]];
l = sold1[run];
m = sold2[run];
n = sold2[run];
pl1 = Plot[sold1[t], {t, 0, 10}, ImageSize -> {200, 200},
  PlotLabel -> "Block 1",
  AxesLabel -> {"time (s)", "displacement (m)"},
  Epilog -> {PointSize[Large], Red, Point[{run, l}]}];

pl2 = Plot[sold2[t], {t, 0, 10}, ImageSize -> {200, 200},
  PlotLabel -> "Block 2",
  AxesLabel -> {"time (s)", "displacement (m)"},
  Epilog -> {PointSize[Large], Red, Point[{run, m}]}] ;
pl3 = Plot[sold2[t], {t, 0, 10}, ImageSize -> {200, 200},
  PlotLabel -> "Block 3",
  AxesLabel -> {"time (s)", "displacement (m)"},
  Epilog -> {PointSize[Large], Red, Point[{run, n}]}] ;

gr1 = Graphics[{Line[{{-0.5, 0}, {0, 0}, {0, 2}}],
   Rectangle[{-1, -6}, {-0.5, 6}],
   Line[{{3 + l, -2}, {3 + l, 0}, {4.5 + l, 0}}], {Lighter[Brown],
    Rectangle[{4.5 + l, -2}, {8.5 + l, 2}]},
   Line[Table[{3 nx/11 + l*nx/11, 2 Cos[\[Pi]*nx]}, {nx, 0, 11}]],
   Line[{{8.5 + l, 0}, {9 + l, 0}, {9 + l, 2}}],
   Line[Table[{9 + l + 3 nx/11 + m*nx/11, 2 Cos[\[Pi]*nx]}, {nx, 0,
      11}]], Line[{{12 + m + l, -2}, {12 + m + l, 0}, {13.5 + m + l,
      0}}], {Lighter[Orange],
    Rectangle[{13.5 + m + l, -2}, {17.5 + m + l, 2}]},
   Line[{{17.5 + m + l, 0}, {18.5 + m + l, 0}, {18.5 + m + l, 2}}],
   Line[Table[{18.5 + m + l + 3 nx/11 + n*nx/11,
      2 Cos[\[Pi]*nx]}, {nx, 0, 11}]],
   Line[{{21.5 + n + m + l, -2}, {21.5 + n + m + l,
      0}, {23 + n + m + l, 0}}], {Lighter[Green],
    Rectangle[{23 + n + m + l, -2}, {27 + n + m + l, 2}]}},
  PlotRange -> {{-1, 40}, {-6, 6}}, ImageSize -> {500, 500}];

Grid[{{gr1, pl1}, {SpanFromAbove, pl2}, {SpanFromAbove, pl3}},
 Alignment -> Top],
{{disp1, 0, "Initial Disp 1(m)"}, -0.1, 0.1, .05,
 Appearance -> "Labeled"}, {{disp2, 0, "Initial Disp 2(m)"}, -0.1,
 0.1, .05,
 Appearance -> "Labeled"}, {{disp3, 0, "Initial Disp 3(m)"}, -0.1,
 0.1, .05,
 Appearance -> "Labeled"}, Delimiter, {{m1, 5, "Mass1 (kg)"}, 5,
 10, .05, Appearance -> "Labeled"}, {{m2, 5, "Mass2 (kg)"}, 5,
 10, .05, Appearance -> "Labeled"}, {{m3, 5, "Mass3 (kg)"}, 5,
 10, .05, Appearance -> "Labeled"}, Delimiter, {{k1, 200,
  "Stiffness 1 (N/m)"}, 200, 750, .05,
 Appearance -> "Labeled"}, {{k2, 200, "Stiffness 2 (N/m)"}, 200,
 750, .05, Appearance -> "Labeled"}, {{k3, 200, "Stiffness 3 (N/m)"},
  200, 750, .05,
```

```
 Appearance -> "Labeled"}, Delimiter, {{run, 0, "release system"}, 0,
  10, .01, ControlType -> Trigger, AnimationRate -> 1}, 
 SaveDefinitions -> True, SynchronousUpdating -> False, 
 AutorunSequencing -> {1, 2, 3}, ControlPlacement -> Left, 
 TrackedSymbols -> True]
```

Coupled Pendulums

```
Manipulate[If[run == 80, run = 0]; L = 9; a = 6; r = 1;
 ic1 = m1*L^2; g = 9.8; ic2 = m2*L^2;
 sol = NDSolve[{{ic1*\[Theta]1''[t] - 
      k*a^2*(\[Theta]2[t] - \[Theta]1[t]) + m1*g*\[Theta]1[t]*L == 0, 
     ic2*\[Theta]2''[t] + k*a^2*(\[Theta]2[t] - \[Theta]1[t]) + 
      m2*g*\[Theta]2[t]*L == 0}, {\[Theta]1[0] == 
     Disp1, \[Theta]2[0] == Disp2, \[Theta]1'[0] == 
     0, \[Theta]2'[0] == 0}}, {\[Theta]1[t], \[Theta]2[
    t], \[Theta]1'[t], \[Theta]2'[t]}, {t, 0, 80}, 
   Method -> "StiffnessSwitching", InterpolationOrder -> All];
 sol1[t_] = \[Theta]1[t] /. sol[[1, 1]];
 sol2[t_] = \[Theta]2[t] /. sol[[1, 2]]; d = sol1[run]; e = sol2[run];
 pl1 = Plot[sol1[t], {t, 0, 80}, 
   PlotLabel -> "Displacement of pendulum 1 ", 
   ImageSize -> {250, 250}, PlotRange -> All, 
   AxesLabel -> {"Time (s)", "Displacement (m)" }, 
   Epilog -> {PointSize[Large], Red, Point[{run, d}]}] ;
 pl2 = Plot[sol2[t], {t, 0, 80}, 
   PlotLabel -> "Displacement of pendulum 2 ", 
   ImageSize -> {250, 250}, PlotRange -> All, 
   AxesLabel -> {"Time (s)", "Displacement (m)" }, 
   Epilog -> {PointSize[Large], Red, Point[{run, e}]}];
 gr1 = Graphics[{{{Lighter[Brown], Rectangle[{-8, 6}, {12, 7}]}}, 
    Black, 
    Line[{{-0.5, 6}, {-0.5 + L*Sin[d/11], 6 - L*Cos[d/11]}}], 
    {{Lighter[Orange], 
      Disk[{-0.5 + L*Sin[d/11], 6 - L*Cos[d/11]}, .5]}}, 
    Line[{{-0.5 + a*Sin[d/11], 6 - a*Cos[d/11]}, {a*Sin[d/11], 
       6 - a*Cos[d/11]}, {1.01*a*Sin[d/11], 7 - a*Cos[d/11]}}], 
    Line[Table[{(3 nx/11 + e nx/22 - d nx/21.75) + 
        a*Sin[d/11], (Cos[\[Pi]*nx])}, {nx, 0, 11}]], 
    Line[{{2.9 + a*Sin[e/11], 4.72 - a*Cos[e/11]}, {3 + a*Sin[e/11], 
       6 - a*Cos[e/11]}, {4 + a*Sin[e/11], 6 - a*Cos[e/11]}}], 
    Line[{{4, 6}, {4 + L*Sin[e/11], 6 - L*Cos[e/11]}}], 
    {{Lighter[Blue], Disk[{4 + L*Sin[e/11], 6 - L*Cos[e/11]}, .5]}}}, 
   PlotRange -> {{-8, 12}, {-5, 12}}, ImageSize -> {400, 300}];
 Grid[{{gr1, pl1}, {SpanFromAbove, pl2}}, Alignment -> Top],

 {{k, 0, "Stiffness (N/m)"}, 0, 10, 0.1, 
  Appearance -> "Labeled"}, {{m1, 1, "Mass of Pen1(kg)"}, 1, 20, 0.1, 
  Appearance -> "Labeled"}, {{m2, 1, "Mass of Pen2 (kg)"}, 1, 20, 0.1,
   Appearance -> "Labeled"}, {{Disp1, 0, 
   "Displacement of Pendulum 1 (m)"}, -3.14, 3.14, 0.02, 
```

```
 Appearance -> "Labeled"}, {{Disp2, 0,
  "Displacement of Pendulum 2 (m)"}, -3.14, 3.14, 0.01,
 Appearance -> "Labeled"}, {{run, 0, "release system"}, 0, 80, 0.001,
  ControlType -> Trigger, AnimationRate -> 1},
SaveDefinitions -> True, SynchronousUpdating -> False,
AutorunSequencing -> {1, 2}, ControlPlacement -> Left,
TrackedSymbols -> True]
```

Quarter Car Model

```
Manipulate[If[run == 20, run = 0]; u[t_] = Sin[\[CapitalOmega] t];
 m1 = 2000; m2 = 250; k1 = 80000; k2 = 800000; b = 10000;
 solu = NDSolve[{{m1*y''[t] + b*(y'[t] - x'[t]) + k1*(y[t] - x[t]) ==
      0, m2*x''[t] + k1*(x[t] - y[t]) + b*(x'[t] - y'[t]) +
       k2*(x[t] - u[t]) == 0}, {y[0] == 0, x[0] == 0, y'[0] == 0,
     x'[0] == 0}}, {y[t], x[t], y'[t], y''[t]}, {t, 0, 20},
   Method -> "StiffnessSwitching"];
 sold1[t_] = y[t] /. solu[[1, 1]];
 sold2[t_] = x[t] /. solu[[1, 2]];
 sold3[t_] = y''[t] /. solu[[1, 4]];
 q = sold3[run];
 d = sold1[run]; f = sold2[run]; g = u[run];
 plot1 = Plot[sold3[t], {t, 0, 20},
   PlotLabel -> "Acceleration of Body -> y", ImageSize -> {200, 200},
   PlotRange -> All,
   AxesLabel -> {"Time (s)",
     "Acceleration (m/\!\(\*SuperscriptBox[\(s\), \(2\)]\))"},
   Epilog -> {PointSize[Large], Red, Point[{run, q}]}] ;
 p2 = Plot[sold2[t], {t, 0, 20},
   PlotLabel -> "Displacement of Wheel -> z", ImageSize -> {200, 200},
    PlotRange -> All, AxesLabel -> {"Time (s)", "Displacement (m)"},
   Epilog -> {PointSize[Large], Blue, Point[{run, f}]}] ;
 p3 = Plot[u[t], {t, 0, 20},
   PlotLabel -> "Displacement Caused by Road -> u",
   ImageSize -> {200, 200}, PlotRange -> All,
   AxesLabel -> {"Time (s)", "Displacement (m)"},
   Epilog -> {PointSize[Large], Green, Point[{run, g}]}] ;
 g1 = Graphics[{
    Line[{{-4, -2 + g}, {8, -2 + g}}],
      Disk[{0, -1.5 + g}, .5],
    Line[{{0, g}, {0, -1 + g}}], Line[{{0, g}, {-1, g}}],
    Line[Table[{Sin[n*Pi - Pi/2], (3*n/10 + f*n/10 + g)}, {n, 0,
       10}]], Text[
     "\!\(\*SubscriptBox[\(K\), \(\(\\\ \)\(US\)\)]\)", {-3,
      2 + f + g}], Line[{{0, 3 + f + g}, {-1, 3 + f + g}}],
    Line[{{0, 3 + f + g}, {0, 4.5 + f + g}}], {{Lighter[Brown],
      Rectangle[{-2, 4.5 + f + g}, {4, 7.5 + f + g}]},
     Text["\!\(\*SubscriptBox[\(M\), \(\(\\\ \)\(US\)\)]\)", {1, (6 +
```

```
      f + g)}]}, Line[{{0, 7.5 + f + g}, {0, 8.5 + f + g}}],
    Line[{{0, 8.5 + f + g}, {-1, 8.5 + f + g}}],
    Line[Table[{Sin[n*Pi - Pi/2], (3*n)/10 +
       d*n/10 + (8.5 + f + g)}, {n, 0, 10}]],
    Text["\!\(\*SubscriptBox[\(K\), \(\(\\\ \\\ \)\(S\)\)]\)", {-3,
      10 + f + g + d}],
    Line[{{0, 11.5 + f + g + d}, {-1, 11.5 + f + g + d}}],
    Line[{{0, 11.5 + f + g + d}, {0, 13 + f + g + d}}],
    {{Lighter[Orange],
      Rectangle[{-4, 13 + f + g + d}, {6, 19 + f + g + d}]},
     Text["\!\(\*SubscriptBox[\(M\), \(\(\\\ \)\(S\)\)]\)", {1, (15.5 \
+ f + g + d)}]}, Line[{{3.5, 7.5 + f + g}, {3.5, 10.5 + f + g}}],
    Line[{{3, 11.5 + f + g}, {3, 10.5 + f + g}, {4, 10.5 + f + g}, {4,
       11.5 + f + g}}],
    Line[{{3.5, 13 + f + g + d}, {3.5, 11 + f + g}, {3,
      11 + f + g}, {4, 11 + f + g}}],
    Text["\!\(\*SubscriptBox[\(B\), \(\(\\\ \)\(OPT\)\)]\)", {6,
      11 + f + g + d}]}, PlotRange -> {{-6, 8}, {-6, 25}},
   ImageSize -> {500, 500}
   ]; Grid[{{g1, plot1}, {SpanFromAbove, p2}, {SpanFromAbove, p3}},
  Alignment -> Top], {{\[CapitalOmega], 0, "Exitation Fr (rad/s)"}, 0,
   5, Appearance -> "Labeled"}, Delimiter, {{run, 0,
   "release system"}, 0, 20, .01, ControlType -> Trigger,
  AnimationRate -> 1}, SaveDefinitions -> True,
 SynchronousUpdating -> False, AutorunSequencing -> {1, 2},
 ControlPlacement -> Left, TrackedSymbols -> True ]
```

Vibration Absorber Model

```
Manipulate[If[run == 10, run = 0]; mr = 0.2; m2 = m1*mr;
 pp = k1/m1; \[Omega]p = Sqrt[pp]; aa = k2/m2; \[Omega]a = Sqrt[aa];
 ra = (\[CapitalOmega]/\[Omega]a); rp = (\[CapitalOmega]/\[Omega]p);

 sol = NDSolve[{{m1*r1''[t] + (k1 + k2)*r1[t] - k2*r2[t] ==
      F Sin[\[CapitalOmega] t],
     m2*r2''[t] - k2*r1[t] + k2*r2[t] == 0}, {r1[0] == 0, r2[0] == 0,
     r1'[0] == 0, r2'[0] == 0}}, {r1[t], r2[t], r1'[t], r2'[t]}, {t,
    0, 10}];
 sold1[t_] = r1[t] /. sol[[1, 1]];
 sold2[t_] = r2[t] /. sol[[1, 2]];
 l = sold1[run];
 m = sold2[run];
 pl1 = Plot[sold1[t], {t, 0, 10}, ImageSize -> {200, 200},
   PlotLabel -> "Main Mass",
   AxesLabel -> {"time (s)", "displacement (m)"},
   Epilog -> {PointSize[Large], Red, Point[{run, l}]}];

 pl2 = Plot[sold2[t], {t, 0, 10}, ImageSize -> {200, 200},
   PlotLabel -> "Absorber",
   AxesLabel -> {"time (s)", "displacement (m)"},
   Epilog -> {PointSize[Large], Red, Point[{run, m}]}] ;
```

```
pl3 = Plot[((1 -
     ra^2)/((1 + mr*(\[Omega]a/\[Omega]p)^2 - rp^2)*(1 - ra^2) -
    mr*(\[Omega]a/\[Omega]p)^2)), {ra, 0, 2},
  AxesLabel -> {"\[CapitalOmega]/\!\(\*SubscriptBox[\(\[Omega]\), \
\(a\)]\)",
    "\!\(\*SubscriptBox[\(U\), \(1\)]\)/\!\(\*SubscriptBox[\(u\), \
\(st\)]\)"}, ImageSize -> {200, 200}, PlotRange -> {{0, 2}, {-4, 4}}];


gr1 = Graphics[{Line[{{-0.5, 0}, {0, 0}, {0, 2}}],
   Rectangle[{-1, -6}, {-0.5, 6}],
   Line[{{3 + l, -2}, {3 + l, 0}, {4.5 + l, 0}}], {Lighter[Brown],
    Rectangle[{4.5 + l, -2}, {8.5 + l, 2}]},
   Line[Table[{3 nx/11 + l*nx/11, 2 Cos[\[Pi]*nx]}, {nx, 0, 11}]],
   Line[{{8.5 + l, 0}, {9 + l, 0}, {9 + l, 2}}],
   Line[Table[{9 + l + 3 nx/11 + m*nx/11, 2 Cos[\[Pi]*nx]}, {nx, 0,
      11}]], Line[{{12 + m + l, -2}, {12 + m + l, 0}, {13.5 + m + l,
      0}}], {Lighter[Orange],
    Rectangle[{12.5 + m + l, -1}, {14.5 + m + l, 1}]},
   Text["k1/m1 =" , {8, 22}], Text[pp , {11, 22}],
   Text["k2/m2 =", {8, 18}], Text[aa, {11, 18}],
   Text["Frq of absorber =" , {9, 14}], Text[ \[Omega]a, {14, 14}]},
  PlotRange -> {{-1, 30}, {-6, 25}}, ImageSize -> {400, 400}];

Grid[{{gr1, pl1}, {SpanFromAbove, pl2}, {SpanFromAbove, pl3}},
 Alignment -> Top],

{{F, 1, "Amplitude (N)"}, 0, 100, .05,
 Appearance -> "Labeled"}, {{\[CapitalOmega], 1,
  "Excitation Frequency (rad/s) "}, 1, 15, .05,
 Appearance -> "Labeled"}, Delimiter, {{m1, 1, "Mass1 (kg)"}, 1,
 10, .05, Appearance -> "Labeled"}, {{k1, 200,
  "Stiffness System (N/m)"}, 200, 750, .05,
 Appearance -> "Labeled"}, {{k2, 1, "Stiffness Absorb (N/m)"}, 1,
 26, .05, Appearance -> "Labeled"}, Delimiter, {{run, 0,
  "release system"}, 0, 10, .01, ControlType -> Trigger,
 AnimationRate -> 1}, SaveDefinitions -> True,
SynchronousUpdating -> False, AutorunSequencing -> {1, 2},
ControlPlacement -> Left, TrackedSymbols -> True]
```

.

REFERENCES

[1] Roy R. Craig Jr. and Andrew J. Kurdila *'Fundamentals of Structural Dynamics'* Second Edition Wiley Publications

[2]*'Estimation of Disturbance Input to a Tire Coupled Quater Car Suspension Test Rig'* Thesis

by Jonathan D. Zeigenmeyer Virginia Polytechnic Institute & State University 2007

[3] *'Model Complexity Requirements In Design Of Half Car Active Suspension Controllers'* Keith J. Wakeham and D. Geoff Rideout, Memorial University, Department of Engineering and Applied Sciences, Canada

[4] *'Study of Transverse Vibration of a Beam With Different Lenght'* Sabah Mohammed Jamel Ali and Zaid Shakeeb Al-Sarraf, Mosul University February 2009

[5] *'Vibration Handout'* Dr. Hanz Richter Cleveland State University

[6] Singiresu S. Rao *'Mechanical Vibrations'* Fourth Edition Pearson Publications

[7] '*Popular Mechanics*' May 2005

[8] *www.wolframalpha.com*

[9] *www.wikipedia.com*

[10] *www.AzimaDLI.com*

BIOGRAPHICAL INFORMATION

Shiva Ramesh Deshmukh received his Bachelor of Engineering in Mechanical Engineering from Nagpur University, Nagpur, India in 2009. He did a voluntary internship at Central Institute for Cotton Research Nagpur before joining UT Arlington in Spring 2010 and started working under Dr. Kent L. Lawrence in Spring 2011. His research interests also include finite element analysis, structural analysis and design. He received his Master of Science degree in Mechanical Engineering from the University of Texas at Arlington in Spring 2012.